Secure Vehicular Communication Systems: Design and Implementation of a Vehicular PKI (VPKI)

Providing Credential Management in a Secure VANET

MOHAMMAD KHODAEI

KTH Electrical Engineering

Master's Degree Project
Stockholm, Sweden November 2012

XR-EE-LCN 2012:015

KTH Electrical Engineering

Secure Vehicular Communication Systems: Design and Implementation of a Vehicular PKI (VPKI)

Providing Credential Management in a Secure VANET

MOHAMMAD KHODAEI

Stockholm Nov. 2012

Lab of Communication Networks (LCN)
School of Electrical Engineering
Kungliga Tekniska Högskolan (KTH)
Stockholm, Sweden

Secure Vehicular Communication Systems:
Design and Implementation of a Vehicular PKI (VPKI)
Providing Credential Management in a Secure VANET

Mohammad Khodaei

Master's Thesis in Computer Science (30 ECTS credits)

at the Information and Communication Systems Security (ICSS)
Master's Program
Royal Institute of Technology, year 2009
Supervisor at School of Electrical Engineering at KTH:
Prof. Panagiotis Papadimitratos
Examiner: *Prof. Panagiotis Papadimitratos*

XR-EE-LCN 2012:015

Royal Institute of Technology
School of Electrical Engineering
Osquldas väg 10
SE-100 44 Stockholm, Sweden
URL: http://www.kth.se/ees

Acknowledgement

Jag kom till Sverige i augusti 2009. Jag har studerat vid *KTH* från och med september 2009 till september 2012. Jag har haft mycket intressanta och roliga tider i Sverige och jag har lärt mig mycket. Mitt program är informationssäkerhet och det kallas för ICSS (Information, Kommunikation och Systemssäkerhet). Under min utbildning har jag haft många utbildade, erfarna och professionella lärare, både från industin och akademin. Dessa har lärt mig väldigt mycket. Jag började också läsa en svenskskurs på *KTH* i juli 2011. Senare fortsatte jag en svenskurs på SFI, två månader i Kistasfilial och ett år på ABFsfilial i Rådmansgatan.

Jag vill tacka för stödet från min handledare, *Prof. Panos Papadimitratos*, som har hjälpt mig med min avhandling och alltid väglett mig när jag stött på olika typer av problem. Han vägledde mig under hela mitt projekt. Jag är uppriktigt tacksam för alla intressanta diskussioner vi har haft. Dessutom skulle jag vilja säga tack till min programdirektor, *Prof. Louise Yngström*, som har hjälpt mig många gånger under min utbildning. När jag har haft problem med min utbildnig har jag kunnat gåtill henne för konsultering. Jag skulle även vilja tacka alla mina uppskattade lärare som jag har haft i Sverige. Jag vill rikta ett särskilt tack till *Prof. Stewart Kowalski*, *Prof. Oliver Popov*, *Prof. Sead Muftic*, *Alan Davidson*, *Prof. Vladimir Vlassov*, *Sarunas Girdzijauskas*, *Maria Wermelin*, *Per Erik Elnerud* och *Johan Montelius*. Tack så mycket för möjligheten att ha fått studera i det bästa landet, Sverige. Tack till alla vänner jag har lärt känna från olika länder och kulturer.

För övrigt uppskattar jag *Nikolaos Alexiou* och *Stylianos Gisdakis* insiktsfulla kommentarer på det tidiga utkastet av rapporten, och de många användbara diskussioner vi hade. Deras råd har varit hjälpsamma under designen av projektet. Dessutom vill jag tacka *Marcello Lagana* för hans tid och hjälp när jag hade problem med att implementera och att installationera OpenCA, och kö ra programmet påservrarna. Han har hjälpt mig en hel del och har bidragit till att spara mycket av min tid. Tack för deras tid, insatser och all jättebra feedback som de gav mig och som förbättrade mina resultat och min avhandlingsrapport. Tack till min vän *Mani Amouzadeh* för alla intressanta diskussioner som vi har haft, och för att du redan från början av projektet har bidragit med synpunkter och kommentarer. Jag vill även tacka skolan för elektro och systemteknik på *KTH* för att ha givit mig tillräcklig utrustning och material för att kunna slutföra min uppsats.

Till sist skulle jag vilja tacka mina föräldrar, *Reza* och *Zohreh*, från djupet av mitt hjärta vill jag tillägna dem min kärleksfulla uppskattning. De har alltid varit med mig i alla situationer, de har stöttat mig i tre år i varje aspekt. Särskilt min syster, *Hoori*, som har uppmuntrat och motiverat mig att fortsätta min utbildning och min älskade systerdotter, *Hasti*.

Jag skulle vilja säga att jag har haft en mycket bra tid i Sverige. Jag har lärt mig många saker under min tid i Sverige. Jag tackar alla lärare som har jag haft i Stockholm. Tack till svenskarna och regeringen som också har låtit mig plugga och fortsätta min utbildning här i Sverige. De har lärt mig mycket om akademin, livet, kultor, religion, sociala system, och mycket annat. Jag kommer aldrig att glöma Sverige och svenskarna. Sverige och det svenska folket har tagit en del av mitt hjärta och jag älskar dem. Jag kommer alltid att vara ett stort Sverige-fan och svenskar-fan. Jag kommer alltid att komma ihåg min vistelse i Sverige och bland svenskarna som en av de bästa tiderna i mitt liv.

Foreword

The master thesis at hand was written during the period of winter 2011 until fall 2012, under the supervision of *Prof. Panos Papadimitratos* in LCN (Lab of Communication Networks) lab in the school of Electrical Engineering at KTH. I am personally indebted to my supervisor, who was always supporting me during the project. I am grateful to work in his team around 1 year while he was kindly answering to all my questions during the meetings. He has guided me all the way from the beginning towards the end.

Thanks to the Ph.D. students who helped me to understand the problems better due to the discussions we had. I would like to appreciate *Nikolaos Alexiou* and *Stylianos Gisdakis* for the excellent feedback and comments on the early versions of the design. Moreover, I would like to appreciate *Marcello Lagana* for all of his time and effort during implementation phase, especially in installing OpenCA as well as integration phase to set up applications on the servers. I would like to thank my friend, *Mani Amouzadeh* for all the interesting discussions we have had from the beginning up to the end of the project. We were keeping in touch while discussing on problems in different phases. Thanks to the department of electrical engineering at *KTH* that provides me sufficient equipments, hardware and software to do my master thesis.

At the end, I would like to appreciate *Prof. Louise Yngström* for all of her supports during 3 years of my education in Sweden. I would also like to dedicate my heartfelt appreciation to my lovely parents, *Reza* and *Zohreh*, my sister, *Hoori*, and my beloved niece, *Hasti*, for all of their supports during my education.

Abstract

The idea of vehicular communication systems could bring more safety, immunity and assurance in driving while it poses a variety of applications in traffic efficiency, driver assistance, environmental hazards, road conditions and infotainment. The aim is to make driving safer and to facilitate driving to the full extent, even on dangerous roads. However, having effective and robust operations within the VC system needs an infrastructure to handle threats, faults, illegitimate activities and unexpected incidents. Message authentication, integrity, non-repudiation and privacy within such a system are considered as the most controversial issues from security perspective. The idea is to protect privacy not only from legal point of view, but also from technical perspective in terms of using privacy enhancing technologies. To provide security within such a system, the idea of Public Key Infrastructure is considered as a promising solution.

Using long-term certificates does reveal the real identity of the owner. Since users' privacy is considered as the main security requirement in the VC system, standard certificates (X.509) and normal PKI cannot be used within a VC network. There are some functionalities and features for vehicular communication systems that do not exist in standard PKI. As a result, using pseudonym certificates to perform transactions within the VC system is a solution.

In this report, a vehicular public key infrastructure, called VPKI, is proposed. OpenCA is used as the PKI, equipped with Pseudonym Certificate Authority (PCA), Long-Term Certificate Authority (LTCA) and Pseudonym Resolution Authority (PRA). These authorities are certified by the RCA and they have privileges to perform their tasks. LTCA is responsible for issuing long-term certificates while PCA is responsible for issuing pseudonym certificates. PRA is the authority to perform pseudonym resolution to identify the real identity of a pseudonym certificate. When it comes to CRL, PCA is the responsible authority to determine revoked pseudonym certificates in order to keep the system secure. Three protocols are then proposed to obtain pseudonym certificates, latest version of pseudonym CRL as well as performing pseudonym resolution. Obtaining pseudonym certificates is done in two phases. Firstly, each vehicle sends a request to LTCA to get a valid token. In the second step, the token is used by PCA to issue pseudonym certificates.

Keywords

Vehicular PKI, Pseudonym CA (PCA), Long-Term CA (LTCA), Pseudonym Resolution Authority (PRA), Pseudonym Certificate, Short-Term Certificate, Pseudonym CRL, Pseudonym Resolution.

Contents

List of Figures

List of Tables

List of Notations

$\sigma_{LTCA}(K_V)$	An LTCA signature on vehicle's long-term certificate
$\sigma_{LTCA}(m)$	An LTCA signature on message m
$\sigma_{PCA}(K_V^i)$	A PCA signature on the public key K_V^i
$\sigma_{PCA}(m)$	A PCA signature on message m
$\Sigma_V^{K_v^i}$	A finite set of pseudonymous public keys for vehicle V
$\sigma_V^{k_v^i}(m)$	V's signature under its i-th pseudonym on message m
$\sigma_V^{LTC}(m)$	V's signature under its long-term private key on m
τ	A period of time
$Cert_{PCA}(k_v^i)$	The pseudonym certificate, signed by a PCA
CRL_{LTCA}	The most up-to-date CRL for long-term certificates
CRL_{PCA}	The most up-to-date CRL for pseudonym certificates
IK	Integrity key, used for SSL secure channel
K_{LTCA}	LTCA's public key
k_{LTCA}	LTCA's private key
$K_{LTCA}(m)$	Encrypt message m with LTCA's public key
K_{PCA}	PCA's public key
k_{PCA}	PCA's private key
$K_{PCA}(m)$	Encrypt message m with PCA's public key
K_V	Vehicle's public key
k_V	Vehicle's private key
K_V^i	i-th pseudonym public key for vehicle V
k_V^i	i-th pseudonym private key for vehicle V
m	Message Payload

$SK(m)$	Encrypt a message using the shared Session-Key
V or Veh	Vehicle V
CA	Certificate Authority
CRL	Certificate Revocation List
EMD	Error Message Description
Err. Code	Error Code
ETSI	European Telecommunications Standards Institute
Exp. Date	Expiry Date
FC	Foreigner Certificate
GPS	Global Positioning System
HSM	Hardware Security Module
IEEE 1609.2	Standard for Wireless Access in Vehicular Environments (WAVE), Security Services for Applications and Management Messages
ITS	Intelligent Transport Systems
ITS-S	ITS Station (either RSU or Vehicle)
LTC	Long-Term Certificate
LTCA	Long-Term Certificate Authority
OBU	On-Board Unit
PCA	Pseudonymous Certificate Authority
PKCS	Public-Key Cryptography Standards
PSID	Provider Service Identifier
RCA	Root Certificate Authority
RSU	Road-Side Unit
ST/LT	Start Time/Life Time
SVC	Secure Vehicular Communication
TCs	Trusted Components
TS	Time-Stamp
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle

VANET	Vehicular Ad hoc Network
VC	Vehicular Communication
VLTC	Vehicle's Long-Term Certificate
VPKI	Vehicular Public Key Infrastructure

Chapter 1

Preliminaries

1.1 Introduction

To the best my recollection, I used to travel with my parents within my homeland, Iran, to many different places where even finding a petrol station was problematic. Exploring the nature in some small villages, with few facilities, was really interesting, amazing and wonderful while it had its own problems. I remember sometimes the car was running out of petrol and my father used to stand outside of the car, in a terrible, cold, snowy or foggy weather, to ask someone help him for a liter of petrol. The idea of vehicular communication systems could make the transportation safer and more efficient and pose a variety of applications in traffic efficiency, driver assistance, environmental hazards, road conditions, infotainment and many other issues. It aims at making driving safer and facilitating driving in dangerous roads all over the world. In this case, maybe no one stands outside of the car for a liter of petrol and hopefully, no one is injured on the roads.

After toll collection or active road-signs vehicular technologies, vehicular communication (VC) systems emerged [1]. The VC system consists of different entities such as: vehicles as well as road-side infrastructure units (RSUs). The entities and units within the system are fitted with *"on-board sensory, processing and wireless communication modules"* [1]. They are communicating with each other using vehicle-to-vehicle (V2V) or vehicle-to-infrastructure (V2I) mechanisms.

The idea of vehicular communication systems can be useful in different circumstances. It goes without saying that in different situations, vehicles need to communicate with each other to disseminate some information. A heavy traffic jam may stem from a simple accident on a highway or it can be due to the overused of resources [2]. Having many cars on the road could result in overwhelming the roadway and a gigantic traffic jam. Without further information about the road condition and the traffic, more cars join the road and subsequently, the traffic gets worse. Furthermore, the lack of disseminating information about the road condition creates a serious problem for an ambulance to make its way through heavy congestion. Dissemination of information about the highway condition, accidents and roadway overused, can be useful to refrain from such problems in advance [2]. The aim is at proposing a new system so that vehicles can communicate and share

the knowledge of the environment with each other. They might propagate warnings if an accident happens, or disseminate the road hazards. As a consequence of this, vehicles are able to choose the best path on the spot, or change their way in the case of an accident and a traffic jam. Providing the vehicular communication system with such facilities can help the transportation system to be used in a more efficient way as it is now.

This chapter begins with a brief overview of vehicular communication systems. Afterwards, key concepts and main entities within the domain of a VC system are addressed. Moreover, three different pseudonym-based authentication schemes are explained. Next, problem statement and research question are pinpointed to determine the current practical problem. The contribution section explains the purpose of this research study, the goal to reach as well as the applied methodology to do the project. This chapter ends with a brief conversation to the audience to point out the target readers as well as limitations and thesis structure.

1.2 Background

A vehicular communication system, as a new generation of distributed system networks, has many differences in compare with other networks and devices connected to the Internet. Some of the main and significant differences, in terms of devices and entities within the system, are explained in [3]:

- Since vehicles have a longer life span in compare with other devices, it is not easy to change or upgrade the on-board system of the vehicles in the case of a new risk. To safeguard the system, the involved entities (such as on-board devices) have to be designed and utilized so that they can tolerate against upcoming risks [3].
- Since the owners of the vehicles have full control over the devices, tampering the hardware and on-board system units is at stake. Owners might try to modify or tamper with vehicles, which cannot be neglected from manufacturer's perspective. Consideration has to be taken when it comes to improve security [3].
- The security of the vehicles has to be completely independent of their owner's reaction. Proper and appropriate security measures have to be put in place to mitigate the threats regardless of the owner's feedback or intervention [3].
- The reactions and alerts to the risks have to be real-time since delays or errors could cause serious problems to human safety or damage and injure passengers. Furthermore, *"robustness requirements and time constraints"* are necessary [3].
- From the legal aspect, there are different regulations in different countries. Reaching a consensus in driving laws, liability and conformance are the most challenging and controversial issues [3].

A VC system is provided with a rich set of tools and applications while enormous amount of attacks threaten the system security. Security within the domain of a VC system is much more of importance since it directly affects the human safety, physical damages and injuries. Many different abuses are possible within the domain of a VC system. Suppose an attacker disseminates enormous amount of false information throughout the vehicular network and subsequently, he contaminates large portions of the VC network. Such faked information could be about road hazards, traffic jam, accidents and weather atmosphere conditions. As an example, an attacker can impersonate an

ambulance to mislead other vehicles to make its way through the traffic jam. All of them results in misleading the normal and legitimate vehicles while it could jeopardize stakeholders within the VC system. A single compromised vehicle can impersonate another entity with higher privileges, such as an emergency vehicle. As a result of this, it can make other vehicles slowed down and thereby, it can take an advantage of that. Another kind of exploit is to infringe driver's privacy by tracking the vehicle, recording some private information regarding the visited places, storing geographical positions and the transactions of the vehicle. Thus, an eavesdropper can infer some private information from the captured data [1].

Due to the fact that the life cycle of vehicles is pretty long, a comprehensive security approach has to be proposed in order to make such a system secure. Many threats exist nowadays and the number of threats increases everyday. To mitigate the threats, a component-based security architecture for vehicular communication system is proposed in [3]. Using such a flexible architecture, components can be easily reconfigured, algorithms can be substituted and more components can be added afterwards [3].

Another important issue to consider is the large number and the variety of vehicles. The ultimate secure VC system has to be proposed in a way so that it can be integrated into all the platforms [3]. In other words, to thwart the existing threats in a VC system, appropriate security measures have to be put in place so that it covers all the platforms; moreover, there is no need to change or upgrade the entire communication system. A hooking architecture is introduced in [3], considered as an event-callback mechanism. This is a flexible interface between different layers of the vehicular communication system [3].

One of the significant security features in the area of a VC system is that security has to be real-time or near real-time [3]. The applied cryptographic hardware guarantees the performance of the algorithms to be real-time. Based on the principle of psychological acceptability[1], adding security to the VC system should not affect the performance of the system as a whole. In other words, the security of the system and its performance have to be kept balanced.

To safeguard the VC system against unknown threats and upcoming attacks, fundamental countermeasures are needed. As elaborated in [3], besides the *"traditional prevention-oriented approach"*, some countermeasures have to be put in place in order to detect malevolent nodes, like: intrusion detection systems. In other words, appropriate mechanisms such as: preemption, deterrence, prevention, detection, deflection, response and recovery are needed against intrusion attempts. The final goal is to *"enhance the resilience of the system"* in the long run [3].

Having successful and perfectly running applications within an infrastructure requires considering some of the outstanding aspects of security. Without considering security in any IT communication system, the system will be vulnerable to attacks, malicious activities and malfunctioning. Many threats are nearby which threaten the system security and as a result, suitable and appropriate measures are necessary to mitigate such threats. Moreover, the aim is to detect and counter attacks on the services running within the system like providing integrity, availability and privacy in Vehicular Ad-hoc Network (VANET) [2].

The security in a VC system has to be built-in, meaning that security shall be considered and applied from the very beginning of the system design instead of adding

[1] According to Bishop: *"The principle of psychological acceptability states that security mechanisms should not make the resource more difficult to access than if the security mechanisms were not present"* [4].

security to the system architecture after designing and implementing the system. Security is not an add-on to each system whereas it is a built-in to every system. Hopefully, industries and academic research centers, specialized within the VC systems, have considered security as a basic requirement from the beginning [2].

There are many attacks which could jeopardize the system performance from security point of view; an adversary model is explained in [1] to identify the attacks applicable to the VC systems from security perspective. Each system is vulnerable to different abuses, threats and attacks and the VC system is not an exception. The details of adversary model is described in the corresponding section, in chapter 2. However, the security requirements within the VC system, are as follows [1]:

- *Message Authentication and Integrity*
- *Message Non-Repudiation*
- *Privacy Protection*
- *Entity Authentication (the aliveness of the sender)*
- *Access Control*
- *Message Confidentiality*
- *Accountability*
- *Liability Identification*
- *Availability*

As provided additional details in [5], the most significant and controversial requirements, in the area of the VC system security, are message authentication, integrity, non-repudiation as well as protecting user's private information, not to allow anyone violating his privacy. The reasons behind considering these four basic requirements as the most critical ones, are as follows:

- *"Authentication and integrity: to prevent message modification and forgery"* [5].
- *"Data consistency: to mitigate the impact of injecting authentic yet falsified measurements"* [5].
- *"Non-repudiation: to prevent nodes from denying transmission of a message"* [5].
- *"Privacy: to prevent the collection or extraction of private information from vehicular communications"* [5].
- *"In-vehicle security: to protect the in-vehicle data access and resources"* [5].

The message confidentiality is not really the most-valuable requirement[2] since the goal is not to keeping messages unreadable from illegitimate entities; but, the must-have properties are transferring messages intact while avoiding entities from message repudiation. On the other hand, asymmetric cryptography provides the requirements to

[2]To clarify more, confidentiality could be an important feature when it comes to protocols like obtaining pseudonym certificates, or long-term certificates. In such protocols, secrecy is critical since some confidential data are exchanging and they have to be kept secret from unauthorized entities. However, the discussion here is about the normal communication in VC system which usually contain traffic and road conditions information, driver assistance, environmental hazards, and infotainment. In such cases, the secrecy is not vital whereas the integrity and non-repudiation are the mostly required services.

authentication, integrity and non-repudiation, as the primitive requirements. Moreover, protecting users' privacy is another necessary requirement and it will be provided by pseudonymity or pseudonymous authentication, which will be discussed in the corresponding section.

The elaboration of the scheme for designing CA for VPKI, Vehicular Public Key Infrastructure, is the main focus of this report. Here, the first version of the VPKI for VC systems has been proposed. Furthermore, the communication protocols among Pseudonymous Certificate Authority (PCA), Long-Term Certificate Authority (LTCA), Pseudonym Resolution Authority (PRA) and Intelligent Transport System Stations (ITS-Ss) have been outlined.

The architectural scheme of CA for VPKI would be designed and deployed using a centralized mechanism while there are some disadvantages in this approach. The upsides and downsides of centralized and decentralized approaches in terms of features, scalability, availability, maintenance, management and problems in implementation are the most controversial issues. For the time being, based on the currently available resources and the source code of OpenCA[3] and Mozilla[4], the dominant solution would be using the centralized-oriented approach. For completeness, a brief comparison between these two approaches is noted here. Using a centralized approach has the advantages such as: easy management and easy configuration. On the other hand, it has the cons like: low scalability, low fault-tolerant, and single point of failure, which might lead to low availability. Using a non-centralized or distributed approach has the strengths like: high scalability, high availability, high fault-tolerant, and no single point of failure. It has also the following drawbacks: difficulties in management and complexity in configuration.

To provide system security requirements within the domain of VPKI, the idea of short-lived keys and long-term credentials are used [6]. It turns out to utilize the concept of pseudonymity or pseudonymous authentication. As a result, each vehicle is equipped with several pseudonymous certificates, valid for a short period of time. It uses such pseudonymous certificates alternately in a way so that messages cannot be linked and consequently, the real identity of the vehicle cannot be disclosed. In this way, vehicles can communicate in a pseudonymous way [6]. More information will be elaborated in chapter 2 and 3.

1.3 Related Works

When it comes to the vehicular communication systems, it seems that security in such networks is a new research field. However, many efforts have been done regarding the security within the domain of Vehicular Ad-hoc NETworks, called VANETs. The related works consider the VC systems security from academic and industrial perspectives.

1.3.1 Academic Research

In terms of security within the domain of a VC system, many efforts have been done so far. [7] proposes the security requirements in the VC system, and it provides a system model to depict the basic communication and security operational within a VC system.

[3] http://www.openca.org/
[4] http://www.mozilla.org/projects/security/pki/

Furthermore, an adversary model according to the protocols are explained. Moreover, the adversary capabilities in a VC system is pinpointed to identify a variety of different attacks. A brief overview of this model is explained in section 2.6.

Moreover, [8] provides a holistic approach to the security of Vehicle-to-Vehicle (V2V) as well as Vehicle-to-Roadside (V2R) communications. Some of the main concepts of security in a VC system is thoroughly considered such as: *"Identification and Secure Addressing"*, *"Privacy"*, *"Access Control and Enforcement"*, *"Intrusion Detection and Containment"*, and *"Protected Communications"* [8]. Parno and Perrig [9] also discussed several adversary types within the domain of a VC system. A comprehensive explanation of different attacks in such a network as well as the corresponding countermeasures and mechanisms to mitigate the threats are outlined in [9].

Papadimitratos [6] pinpoints the Certificate Revocation List Distribution in VANETs. When it comes to protecting privacy, [10] proposes an efficient and robust pseudonymous authentication in VANET to protect users privacy withing such a network. Such concepts are clarified in section 1.4 in more details. In [1], Papadimitratos also considered the design and architecture of a secure VC system as well as the basic security requirements. Such requirements are message authentication, integrity, non-repudiation as well as protecting users' private information, not allowing anyone violating their privacy. Section 2.4 provides a detailed explanation of security requirements for VANETs.

[11] proposes a conditional pseudonymity approach in VANETs which resolves the pseudonyms without storing information on authorities. Having stored no pseudonym-identity mappings on the authorities, they should cooperate with each other to resolve the pseudonyms. To put it in another way, pseudonym-identity mappings are only stored on the pseudonyms, as addressed in *V-tokens for Conditional Pseudonymity in VANETs*. In this paper, the assumption is that authorities are trusted and they never store resolution information. Moreover, each vehicle signs the pseudonym certificate request using its current valid pseudonym, which results in the linkability problem between different pseudonyms. The resolution information including: id_V, id_{CA} and a unique randomized factor r are encrypted using the corresponding resolution authority's public key.

1.3.2 Industrial Projects

There were also many projects in the area of security in VC systems in Europe including NoW (Network on Wheels)[5], GST (Global System for Telematics)[6] and the Car2Car Communication Consortium (C2C-CC)[7]. The ongoing project in European is SEVECOM (SEcure VEhicular COMmunications)[8] that focuses on providing complete security requirements for the VC systems [5][12]. As explained in [5]:

> "SEVECOM will further investigate vulnerabilities, model attackers, perform a risk analysis, and identify security requirements, in liaison with other related projects, including NoW, C2C-CC, GST, the eSafety projects CVIS, SafeSpot, Coopers, and COMeSafety. The methodology underlying the ISO 15408 standard on security evaluation (common criteria) will be used to carry out the threat and risk analysis. Beyond technical requirements, business and legal constraints and requirements will be taken into consideration."

[5] http://www.network-on-wheels.de/
[6] http://www.ertico.com/gst-website
[7] http://www.car-to-car.org/
[8] http://www.sevecom.org/

PRESERVE [12], mainly focuses on the security and privacy features for V2V as well as V2I communication systems. As stated in [12], performance, scalability, and deployability of V2X security systems are the most controversial and critical issues within this project.

1.4 Definitions and Key Concepts

This section consists of key concepts, terminologies, definitions, vocabularies, and other kinds of conceptual knowledge used within this report. Having known the concepts, the report is more readable and understandable. At first, the role of different authorities, identified as RCA, PCA, LTCA and PRA is explained. Later on, 3 different schemes for pseudonym-based authentication are described. The last scheme, hybrid pseudonym, is used in the communication within the secure VC system.

1.4.1 The Role of RCA, PCA, LTCA and PRA in VPKI

Here, a detailed description of the role of each authority is illustrated. In the domain of secure vehicular communication systems, there exist 4 different authorities, each plays different roles. All of them are assumed to be benign entities and they are considered completely trustworthy third-parties. Figure 1.1 shows the relations between different hierarchies of authorities within the domain of a VPKI. The role of each authority is pointed out below.

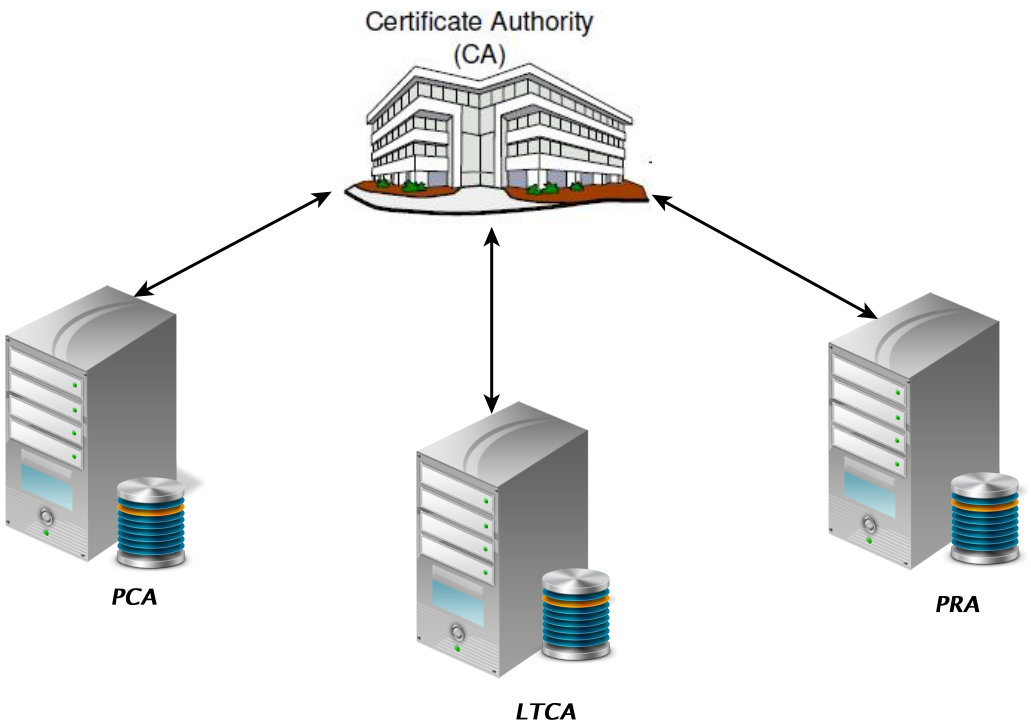

Figure 1.1: Authorities Hierarchy in VPKI

RCA: Root CA

RCA is the trust anchor within VPKI and its certificate is self-signed. In general, each certificate has a field called signer-Id. RCA certificates have no signer-Id since its certificate is signed by itself. The role of RCA in a vehicular communication system is to sign and issue certificates for other authorities like: PCA, LTCA and PRA. It is crystal clear that RCA certificate is publicly available to every vehicles, ITS stations and other authorities. To make it more general, each ITS station has to store the certificates of RCA, PCAs, LTCAs as well as PRA.

PCA: Pseudonymous Certificate Authority [9]

PCA is the entity, issuing pseudonymous certificates for the vehicles. Each vehicle sends a request to an appropriate PCA[10] to achieve a set of pseudonymous certificates. Having had pseudonymous credentials, each vehicle can communicate with other nodes anonymously without being monitored, tracked or identified. The idea here is to make it very difficult to make a link between different messages, signed by the same entity. [11] The details of how to obtain the pseudonymous credentials will be described in details in chapter 3, design protocol.

LTCA: Long-Term Certificate Authority [12]

LTCA is another trusted entity within the domain of a secure VC system to issue long-term certificates for the vehicles, RSUs and other entities. Each entity has to send its request directly to LTCA in order to achieve a long-term certificate. The long-term certificates are used in order to obtain pseudonymous credentials. Moreover, LTCA is responsible to issue tokens for the vehicles, used in the process of obtaining pseudonym certificates. The details are discussed in design protocol, chapter 3.

PRA: Pseudonym Resolution Authority [13]

There is another entity in VPKI, called Pseudonym Resolution Authority (PRA), certified by RCA. The role of PRA is pseudonym resolution. It is able to query a PCA to figure out the related token to a specific pseudonym. Later on, PRA queries LTCA to identify the real identity for that token. It this case, if a pseudonym is identified to be malicious, PRA is capable of revoking all the related pseudonyms accordingly. Further discussion is presented in chapter 3.

1.4.2 Pseudonym-based Authentication Schemes

When it comes to the security services provided within the VC system, message authentication, integrity and non-repudiation are considered as the main and most significant services [10]. However, from the users' standpoint, protecting their privacy and keeping their real identity secret are much more important since they do not want to be tracked, monitored or identified. In other words, cryptographically protected

[9] There is no similar entity to PCA in the standard PKI.

[10] The appropriate PCA implies the closest PCA to the vehicle V.

[11] Since this is not a full-anonymous communication, there is a chance to identify the real identity of vehicles after a period of time. That is why vehicles change their credential once in a while. To be more precise, vehicles communicate in a pseudonymous manner, not anonymously.

[12] LTCA in the domain of a secure VC system is similar to the RA (Registration Authority) in the standard PKI.

[13] There is no similar entity to PRA in the standard PKI.

messages have to be kept anonymous while they have to be identified and authenticated as well. In addition, making a link between two messages generated by the same node shall be difficult for an eavesdropper. To achieve this goal, the basic idea is to change the signing key and the corresponding public key once in a while. Thus, the messages are signed under different keys and as a result, they cannot be linked. Hence, only messages, signed under the same private key or verified under the corresponding public key, can be linked [10].

There are three main approaches regarding pseudonym-based authentication identified as: Baseline Pseudonyms (BP), Group Signatures (GS) and the combination of these two approaches called Hybrid Pseudonym. The details of each scheme are discussed in [10]; however, these schemes are considered as key concepts within the area of secure vehicular communication systems. Hybrid pseudonym will be used for VC systems as the authentication approach. Thus, the basic idea behind each of them is introduced briefly.

Before digging into the explanation of these schemes, it is worth-mentioning that the assumption is that long-term keys are generated, disseminated and perfectly stored on each vehicle; in other words, no faked certificate is stored as a genuine or true credential. The details of protocols on how to acquire pseudonyms will be discussed in chapter 3.

1.4.2.1 Baseline Pseudonyms

In the first scheme, each vehicle comprises several pseudonyms, meaning that several pairs of public key and private key are enclosed with each legitimate vehicle. Each public key is signed by the CA, identified as $Cert_{CA}(K_V^i)$, while it does not reveal the real identity of the vehicle. The set of pseudonymous public keys are identified as: $\Sigma_V^{K_v^i}$. To digitally sign a message, each vehicle uses one of its pseudonymous private key, k_V^i.[14] In order to validate a signature on the receiver side, the pseudonymous public key, K_V^i, as well as the signer's certificate, $Cert_{CA}(K_V^i)$, have to be transmitted in each message. $\sigma_{k_V^i}(m)$ also defines the vehicle's signature using its i-th pseudonym private key, k_V^i [10]. The message format is indicated below:

$$m,\ \sigma_{k_V^i}(m),\ K_V^i,\ Cert_{CA}(K_V^i)$$

Although the real identity of each vehicle is kept secret and vehicles communicate anonymously, the CA keeps a mapping between the set of pseudonyms and the long-term identities of vehicles. This means that in the case of accidents or illegal activities, a trusted authority is able to do investigation and perform forensics techniques in order to find chain of custody. The interesting point from privacy perspective is that each pseudonym is discarded after a period of time τ [10].

The recipient uses the CA's public key to validate $Cert_{CA}(K_V^i)$. Afterwards, it can validate $\sigma_{k_V^i}(m)$ using the validated and authenticated vehicle's certificate. The assumption is that the certificates of the CAs as well as the latest CRLs have been already disseminated to the nodes.

1.4.2.2 Group Signatures

In the second scheme of pseudonym-based authentication, each node V legitimately registered within the domain of a VC system, is equipped with a secret *group signing*

[14] Note that the public key is identified by K_V^i (CAPITAL K) whereas the corresponding private key is indicated by k_V^i (non-capital k).

key, gsk_v. Each node signs messages using gsk_v, without disclosing its real identity; moreover, it can validate signatures using *group public key*, gpk_{CA}. In other words, *group signing key* enables legitimate nodes within the VC system to sign anonymously on behalf of the group without revealing theirs real identities. Besides, making a link between two different signatures to identify a node is impossible. Interestingly, the message below is sent while neither public key, nor other credentials are enclosed [10]:

$$m,\ \Sigma_{CA,V}(m)$$

Furthermore, in the case of investigation, the CA can perform an *Open*[15] operation [13][14] to figure out the signer's identity. In a group signature scheme, a revocation list is also constructed to determine the revoked nodes within the group [10].

1.4.2.3 Hybrid Pseudonym

Hybrid pseudonym, which will be used as the authentication scheme in vehicular communication system, is a combination of the two previous schemes, baseline pseudonyms and group signatures. Each vehicle V is provided with a group signing key, gsk_v, and a group public key, gpk_{CA}. In this scheme, each vehicle V generates its own set of pseudonyms, recognized as $\Sigma_V^{K_v^i}$[16]; the corresponding private keys are identified as $\Sigma_V^{k_v^i}$[17]. It uses gpk_V in order to generate $\Sigma_{CA,V}()$ for each group signature K_V^i [10].

As illustrated in [10], K_V^i is generated and *"self-certified"* by vehicles, producing $Cert_{CA}^H(K_V^i)$[18]. In this case, each vehicle includes $Cert_{CA}^H(K_V^i)$ as well as the signature $\sigma_{k_V^i}(m)$, to each message while it sends to the recipient:

$$m,\ \sigma_{k_V^i}(m),\ K_V^i,\ Cert_{CA}^H(K_V^i)$$

The group signature, $\Sigma_{CA,V}(K_V^i)$, on the recipient side, will be validated by gpk_{CA}. Having verified and validated the signature, the recipient realizes that a legitimate node within the VC system has signed the message; moreover, no entity can link any two messages to identify the identity of the sender. In other words, vehicles managed to communicate with each other in a pseudonymous manner,[19] while the legitimacy of the signer as well as the validation of the signature are totally proved. In the same way, a trusted third-party is able to figure out the real identity of the signer, using an *Open* operation [13][14], in the court of law [10]. As provided with additional details, *"the message m is bound to* K_V^i *via* $\sigma_{k_V^i}(m)$*, and* K_V^i *is bound to V via* $\Sigma_{CA,V}(K_V^i)$*"* [10].

It is worth-mentioning that three optimizations were proposed to reduce the overhead of the message signature and the verification. The reason is to enhance the robustness and

[15] Based on the Separation of Duties (SoD) principle, the authorities are separated into two distinct entities, an opener, able to open signatures, as well as an issuer, the responsible entity to interact with users in order to issue a signing key. With two separated entities, the security of the system is enhanced as a whole. The general idea is to equip an authority with a traceability feature in order to read the registration table, populated by the issuer authority [14]. In other words, open operation and open algorithm provide a mean of full-traceability as well as a strong form of collusion-resistance [13].

[16] CAPITAL K

[17] non-capital k

[18] To differentiate hybrid pseudonym with other schemes, an H superscript is placed. Also, the CA subscript proves that the certificates are legitimately signed by CA.

[19] To be more precise, the communications are not fully-anonymous while they are considered as pseudonymous. Since the cost of having fully-anonymous communication is very high, pseudonymous communication is used because it is not useful to have a completely anonymous communication here.

the performance of the protocols. Although they are not discussed in details here, a brief discuss would be interesting. The optimizations mostly deal with sending the unchanged fields only once throughout the lifetime τ on the sender side. Moreover, verification of the unchanged fields is done only once on the recipient side. Additionally, enclosing a 4-byte *keyID*[20], indicates the corresponding K_V^i to validate the signature $\sigma_{k_V^i}(m)$ on the recipient side.

1.4.2.4 Pseudonym-based Authentication Schemes, A Comparison

Table 1.1 illustrates a comparison among different message formats for different pseudonymous authentication schemes, as described in details above.

Table 1.1: *A Comparison among Pseudonymous Authentication Schemes [15]*

Pseudonymous Authentication Schemes	**Message Formats**
Baseline (BP) Scheme	m, $\sigma_{k_V^i}(m)$, K_V^i, $Cert_{CA}(K_V^i)$
Group Signature (GS) Scheme	m, $\Sigma_{CA,V}(m)$
Hybrid Scheme	m, $\sigma_{k_V^i}(m)$, K_V^i, $Cert_{CA}^H(K_V^i)$

1.5 Problem Statement and Research Question

The problem at hand is the lack of an infrastructure to provide secure communications between different entities within the domain of a vehicular communication system. To secure the communications, entities have to be capable of utilizing and performing security services which requires an infrastructure to be applied. In the nowadays communication schemes, there exists an infrastructure called PKI, public key infrastructure, which provides a platform so that legitimately registered entities within the domain are able to communicate securely and apply security services as well.

In the other words, the current vehicular communication system lacks an infrastructure such as a PKI. Public Key Infrastructure is considered as an essential requirement to provide credential management and security services. It is also a necessary requirement to build an IT system secure. No need to mention that there is no reasonable alternative to achieve such a level of security without a PKI. Thus, the practical problem is a gap between the current state of the system and the ultimate desirable state of the VC system, appropriate for the vehicular communication systems. Having designed and deployed a desirable PKI for the VC system, the entities within the system are capable of performing security services, communicating securely and achieving the basic security requirements.

Based on the standardized entities and algorithms, like CA, RA, digital signature, RSA algorithm, public/private key pairs, and so forth, the idea to design an infrastructure for the VC system is inspired. There are many tools in this area, customized by different companies within several projects. The goal of this project is to build a vehicular public key infrastructure called VPKI so that all the application, which communicates using IEEE standard 1609.2, can be integrated within such a new system called secure VC system.

Above all, building a new public key infrastructure for VC systems deals with many different issues, which must be considered. The idea is to utilize the currently available

[20] As explained in [10], *keyID* is a random number while it does not infringe the privacy at all.

open-source PKI, such as OpenCA or Mozilla, to build such an infrastructure. Thus, the source-code for this purpose is not supposed to be written from the scratch while the idea is to utilize and customize the currently available resources. The security services and entities involved within this infrastructure are as follows:

- *Message Authentication and Integrity*
- *Message Non-Repudiation*
- *Entity Authentication*
- *Access Control*
- *Privacy Protection*
- *Liability Identification*
- *Availability*
- *Certificate Revocation*
- *Key and Identity Management*
- *Secure Communication Protocol*
- *RCA, PCA, LTCA and PRA as the Trusted Third-Parties (TTPs)*
- *Hardware*
- *Software*
- *People (such as: cars, motor cycles, public and private vehicles, RSUs)*

The main focus would be on message authentication, integrity and non-repudiation as well as protecting users' privacy, considered as the most important security requirement. These features and services will be elaborated in the subsequent chapters.

1.6 Contribution

The practical problem at hand is the lack of a secure infrastructure in the VC system. Without a trusted and secure infrastructure, vehicles cannot communicate securely while they are exposed to different threats and attacks. Within such a domain, an adversary can stage an attack, jeopardize users' privacy and disclose confidential information. Thereby, an attacker can exploit the vulnerabilities and as a result, he can violate the VC system security policy. To thwart the threats, safeguard the stakeholders and make the system operation secure, there needs to design and deploy an infrastructure, called Vehicular Public Key Infrastructure.

In the following, the research purpose of this project is stated. Afterwards, the goal of solving such a practical problem is determined. The contribution ends with the research methodology to pinpoint how the existing problem will be solved.

1.6.1 Research Purpose

To the best of current knowledge, there has not been any investigation in implementing a VPKI for a secure VC system so far. This is the first investigation, focused on designing and implementing a VPKI for the secure VC system. The research purpose of this project is planed carefully with great details to fulfill this gap. As a result of this, the VC system is equipped with an infrastructure called VPKI, to enable entities registered within the domain to be interconnected while they communicate securely.

1.6.2 Goal

The goal of the project is to build an artifact, using the currently available open-source PKI, for the secure VC system. In other words, the outcome of this research project will be an implementation of a VPKI. On the way towards implementation, several protocols are needed to be designed. The proposed protocols need to be formally proved, in terms of security. These protocols are mainly dealt with how to achieve pseudonymous credentials, long-term certificates as well as disseminating the latest pseudonym CRL as well as performing pseudonym resolution.

1.6.3 Methodology

When it comes to the methodology of a research project in Information System discipline, there exist two alternatives, namely behavioral science and design-science [16]. Based on the definition of design-science, *"behavioral science paradigm seeks to develop and verify theories that explain or predict human or organizational behavior"* [16]. The design-science paradigm, on the other hand, *"seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts"* [16]. The aim in design-science is to design, deploy and implement a new artifact in order to achieve knowledge, understand the practical problem and propose an appropriate solution to the existing problem [16].

Design-science is considered as a problem solving paradigm, which combines theory and practice to develop an artifact in order to solve a practical problem [16][17]. As stated before, the practical problem in this project is the lack of a secure infrastructure within the domain of vehicular communication systems. The lack of this infrastructure may be attributable to many security exposures. To solve this practical problem, the design-science paradigm is used to create an artifact and thereby, mitigate the security breaches and make the domain of a vehicular communication system secure.

Design-science research approach comprises two design processes, stated as build and evaluate; moreover, it determines four design artifacts which consists of *"constructs (vocabulary and symbols), models (abstractions and representations), methods (algorithms and practices), and instantiations (implemented and prototype systems)"* [16]. Design-science aims at creating and evaluating an artifact to solve an existing practical problem. The idea is to run design-and-evaluate loop several times in order to achieve better feed-backs and design the product to the full-extent of quality [16].

[16] proposes 7 guidelines to the design-science researches. A brief discussion of each guideline is addressed here. *"Design as an Artifact"* states that a *"design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation"* [16]. *"Problem Relevance"* mostly focuses on the objective of a design-science research. *"Design Evaluation"*, as an important phase of design-science research, deals with methods and procedures to rigorously evaluate the quality of the design and subsequently, the quality of deployed artifact. *"Research Contributions"*, considered as the 4th guideline, emphasizes on a clear and verifiable contributions in the design. *"Research Rigor"* addresses the rigorous methods in design, construction as well as evaluation of the artifact. *"Design as a Search Process"* and *"Communication of Research"* explain how to reach the desired ends from technological and management perspectives [16].

In order to overcome the complexity of solving a practical problem, [17] provides different activities as the design-science method, sketched in figure 1.2. In this diagram,

different phases and activities are depicted, called design-science method. These guidelines would entail different ranges of activities from problem investigation, requirements definition, demonstration and evaluation of the artifact. Although the design-science method illustrated sequentially in figure 1.2, it is usually carried out in an iterative manner among problem explication, requirement definition, development as well as evaluation [17].

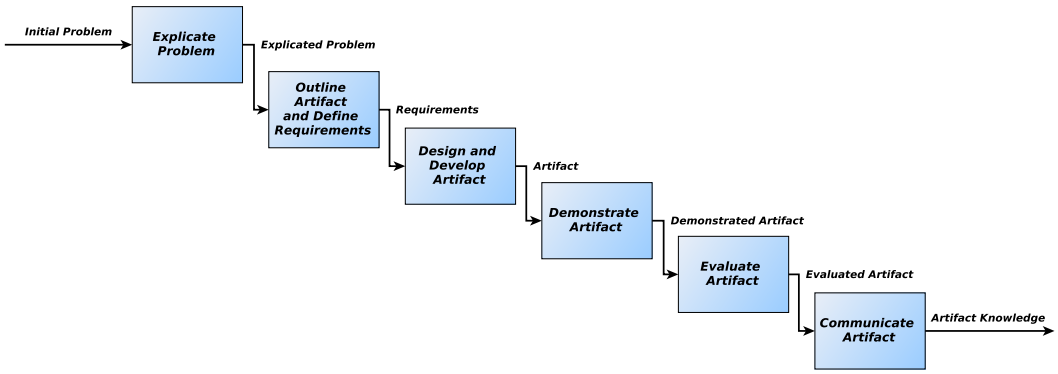

Figure 1.2: Overview of the Design-Science Method (Johannesson & Perjons 2012) [17]

In the next section, the applied research strategy according to the current practical problem is explained. Based on the design-science method, each phase is pointed out. In other words, this research strategy is targeted to explain what is the existing practical problem in the vehicular communication system, why it is important to be solved and how the problem will be solved.

1.6.3.1 Research Strategies

Each of the phases and activities regarding design-science method is pinpointed according to the existing problem. As mentioned earlier, design-science deals with producing an innovative and effective artifact to solve a practical problem. The very beginning phase of design-science method is to explicate the problem; it follows by requirement definition, design and develop, and finally evaluation. In the following, each phase is clarified with additional details.

Explicate Problem

The first phase aimed at pinpointing the practical problem. The practical problem has to be investigated, analyzed, formulated and thoroughly motivated. The reason is to show

that solving this problem has several benefits from different aspects. In the other words, it is significant to evince that solving this problem would fulfill a gap, and move the current undesirable state to a desirable state. Moreover, the idea is to provide evidences that solving this problem is not only of a specific or local interest, but also of a broader and general interest. The research strategy to answer problems and questions regarding the artifact could be within the variety of strategies while it depends on the characteristics of the project, its goal and the environment [17].

To explicate and scrutinize the problem, surveys, interviews with experts in that area, as well as questionnaires are considered as useful and effective instruments. Observations and document studies are also other important tools to achieve knowledge about the practical problem [17]. For the current research project, document studies, literature review, empirical experience and interviews are the mostly useful instruments to achieve knowledge and understand the problem while surveys and questionnaires are of less important. Specifically, interviews with experts were used to elicit requirements and the goals of the project since such information cannot be achieved from other resources.[21]

The current practical problem, which is the main focus of this research study, is the lack of an infrastructure to provide security services. Without such an infrastructure, the communications are not secure while there exist many threats to the system. The importance of making the communications secure in the VC system is due to the physical damages and injuries to the human. An attacker can easily mislead vehicles, inject faked information and as a result, he can create hazards on the road. A detailed discussion on the attacks within the domain of a VC system is explained in chapter 2, the *Adversary Model*. To mitigate the threats, an infrastructure called VPKI is necessary to avoid the attacks. Having designed and deployed such a VPKI, other security services can also be applied and utilized. Since VPKI is a really broad area of research, the main focus of this research study is only on Certificate Authorities, namely RCA, PCA, LTCA and PRA. In other words, the communications between ITS stations with the PCA, LTCA and PRA is a matter of discussion. The details of communication to achieve pseudonym (short-term) certificates[22], pseudonym CRL as well as pseudonym resolution are provided with great details in chapter 3.

Outline Artifact and Define Requirements

The main purpose of this activity is to specify an appropriate solution to the previously explicated problem [17]. Based on the prior step, the requirements are defined and an appropriate artifact will be proposed. In this specific research study, both functional requirements as well as security requirements are identified; followed by evaluating the current requirements and assumptions against the VC adversary model. Finally, appropriate and effective security protocols are proposed in order to achieve long-term credentials, short-term credentials and the CRLs. The problem at hand is to propose security protocols within the VPKI domain so that entities are able to be interconnected securely. The details of such protocols will be illustrated in the appropriate section in

[21] Since this research study is a part of PRESERVE (Preparing Secure Vehicle-to-X Communication Systems) project [12], many discussions have been done with the supervisor; moreover, many brainstorming sessions have been held in order to come up with new ideas. As a result of these discussions and meetings, the requirements for the thesis project were identified. Furthermore, the preceding projects such as Sevecom[5] and many other scientific papers in the area of the VC system were quite useful in order to picture the goal of the thesis project.

[22] Within the domain of a secure VC system, pseudonym certificates and short-term certificates are being used interchangeably.

chapter 3. However, the important factor here is that this step is being done using the analysis of predefined requirements, several scientific and reputable articles as well as interviews and discussions with experts in this area.

Design and Develop Artifact

The third step in design-science method is to design, develop and implement the proposed artifact based on the explicated problems to thoroughly fulfills the requirements [17]. To achieve this, a comprehensive investigation on the currently available open-source PKIs is performed, which will be outlined in chapter 4. Based on that investigation, OpenCA and Mozilla are selected out of 10 proposed open-source PKIs; the criteria of selection are the completeness of documentations, implementation language, and level of support. The most important of all, is the similarities of the required features within the domain of VPKI to the current implementations of the standard PKI. Due to the selected approach, best practices, ETSI[23] documentations and IEEE 1609.2[24] are used.

Demonstrate Artifact

This phase describes the use of developed artifact in the real environment or in other words, in the *"real-life case"* to prove its feasibility [17]. In this case, the final version of the artifact and fully-tested implementation of VPKI shall be integrated into the system in order to demonstrate the implemented features and requirements. The goal is to illustrate that using this artifact, the explicated problem will be solved.

Demonstrating the deployed artifact needs integrating the VPKI into the real environment. To perform this step, the real project[25] has to be first finalized and ready to be demonstrated. Since the project research and deployment are still going on, this needs more time to carry out this phase of design science. Furthermore, not all the features and services of VPKI are expected to be deployed and implemented in this thesis project. Taking all of such limitations into consideration, demonstrating this artifact in the so called *"real-life case"* is not feasible.[26]

Evaluate Artifact

Artifact evaluation deals with identifying and determining whether the artifact is made up of all the necessary requirements, to determine to what extent this approach has solved the problem [17]. In this step, the functional requirements and security requirements have to be evaluated and fully-tested. From the security perspective, security protocols have to be evaluated to the full-extent using formal methods. Any error or fault has to be identified and alleviated.

Formal methods and experiments are the most important instruments to evaluate the artifact and specially, in the current VPKI. The proposed security protocols within the VPKI are evaluated by other experts to identify security breaches. The idea is to carry out several experiments in different conditions and specifically, in the real environment[27] to evaluate the artifact [17].

[23] European Telecommunications Standards Institute

[24] IEEE Trial-Use Standard for Wireless Access in Vehicular Environments-Security Services for Applications and Management Messages, Intelligent Transportation Systems Committee, Sponsored by the IEEE Vehicular Technology Society.

[25] PRESERVE project. More information, online: http://preserve-project.eu/.

[26] The main issue is the limitation of time in order to finish the master thesis.

[27] Just to emphasize once more, the artifact will not be evaluated in the real environment due to the mentioned limitations.

Taking into consideration, a full-scale evaluation of this artifact could not be carried out due to the time and resource limitations. Assessment and validation of the proposed security protocols will be performed to some extent; however, the evaluation strategy in this project is an *"ex-ante evaluation"*. There are mainly two strategies to evaluate: *"ex-ante evaluation"* and *"ex-post evaluation"*. An *"ex-ante evaluation"* is evaluating the artifact without being used [17]. However, an *"ex-post evaluation"* assesses the artifact while it is being employed [17]. In this case, the artifact and security protocols are evaluated by experts. The experts and researchers declare their views and comment on the protocols and the artifact, using their experience in similar projects. The main shortcoming of this approach is that the evaluation is subjective and it cannot provide an effective and perfect results; however, it provides valid and reliable results, to some extent, while it is being done with low cost [17].

Communicate Artifact

The main purpose of the final phase of design-science is to communicate the knowledge, achieved within the process of creating the artifact. The aim of this research study is also to document the research activities and the results to be available to the researchers, experts and practitioners.

1.7 Audience

The report at hand is written to inform experts within the domain of information security, system security, communication and distributed systems. In this report, the principles of computer security, security services, computer networks, communication and distributed systems and specifically, the basic concepts of public key infrastructure are not explained. It is assumed that readers are familiar with these concepts, although a brief description are stated in different parts of the report. Moreover, to get the most out of this thesis report, the readers shall be familiar with UML[28] as well as software programming concepts.

1.8 Limitation

A lot of research and efforts have been conducted on the area of vehicular communication systems. The research project at hand tries to propose a solution to only one of the existing problems within the domain of a secure VC system. It goes without saying that this is only one of the approaches regarding providing credential management to secure a VC system. It is obvious that the proposed approach is one of the most reasonable alternatives while it is considered practical and effective. The limitation here is that the project is supposed to be done within 8-10 months of full-time work. Thus, this project is being done to the full extent that time permits. Furthermore, the aim is at utilizing the free and open-source software due to the limitations with commercial software. Moreover, implementing the whole VPKI is obviously impossible in a master thesis project. The idea is not to implement all the required services and features; whereas, it is expected to implement and deploy the mentioned protocols and evaluate them. It is assumed that this project will be continued by other experts to complete the VPKI in order to be used in practice.

[28] Unified Modeling Language

1.9 Thesis Structure

The report at hand is categorized in different chapters. The first chapter is an introduction to vehicular communication systems. In this chapter the key concepts of vehicular communication system is discussed. Moreover, the research purpose, its goal and the applied methodology are addressed.

The second chapter, VPKI, is dedicated to the general idea and definitions of VPKI. In this chapter, all the notations, which will be used throughout the report, are symbolized and specified. Furthermore, the functional requirements as well as security requirements of the VPKI are declared. The VPKI infrastructure is also proposed; it ends with an adversary model to figure out the existing security breaches and exposures.

Chapter 3 is considered the main contribution of this project when it comes to design. In this chapter, three different protocols on how to achieve pseudonymous certificates, the latest CRLs, as well as pseudonym resolution will be expressed.

Chapter 4 is dedicated to the deployment and the real implementation of the project. In this chapter, a brief comparison of different implementations of open-source PKI will be discussed. The chosen libraries and the details of source code and implementations of VPKI come here.

Chapter 5 gives a detailed description of the evaluation of the work. The security of the designed protocols and the corresponding implementation are also evaluated. The experimentation setup is also sketched, follows by providing additional details on the results of experiments.

Chapter 6 offers the conclusion and the future works. The future works state the remaining tasks to provide more services and improve the features of a VPKI. It helps other researchers in order to pursue this study of research.

An appendix comes at last to demonstrate the details of the protocols line by line. Each protocol is provided with a UML diagram to visualize the protocol procedure.

Chapter 2

Vehicular Public Key Infrastructure

2.1 Introduction

The vehicular public key infrastructure is considered as the best alternative to provide credential management and security services within a VC system. Such an infrastructure is identified as the most critical and vital part of the system since all the security services will be built over it. In other words, it is the base security infrastructure and thereby, it has to be designed, deployed, evaluated and maintained to the full extent, from security point of view. Each secure communication depends on the strength of this infrastructure; a simple vulnerability within the system could jeopardize the security of the entire system. Different aspects of the system and the environment should be considered in order to make such a broad system secure. The assumptions, requirements, existing threats, and relevant adversary models have to be scrutinized so that proper countermeasures could be put in place in advance. Although this is a broad area, many efforts have been done regarding adversary models in a VC system [7] and CRL protocols [6].

In this chapter, the acronyms and notations, used in this research study, are identified. When it comes to explain different protocols in chapter 3, such notations will be used to show the communication among different entities. Afterwards, the assumptions within the domain of a VC system are addressed. These assumptions are very important since the requirements are proposed according to such assumptions. Next, functional requirements and security requirements in a VPKI are mentioned. Later on, the proposed VPKI infrastructure is depicted; the communications among ITS-Ss, a PCA, an LTCA and a PRA are shown in a diagram. This chapter ends with an adversary model to identify the threat and possible attacks within a VC system. Such adversary model has been proposed to mitigate the threats in a vehicular communication system.

Two different symbols are used for signature. The first one is $Cert_{PCA}(m)$, which means the PCA signature, as an authority, on message m. The other one is $\sigma_V^{LTC}(m)$, used to identify the signature of vehicles under their long-term private keys. In other words, the signature of vehicles is identified by σ_V while the signature of other authorities is denoted

by $Cert_{PCA}$ or $Cert_{LTCA}$. To be more clear about the notations, the signature of authorities and vehicles are represented below:

- $Cert_{LTCA}(m)$*: illustrates the LTCA signature on a message m.*
- $Cert_{PCA}(m)$*: specifies the PCA signature on a message m.*
- $Cert_{PCA}(K_V^i)$*: describes the PCA signature on a short-term public key* K_V^i.
- $\sigma_V(m)$*: symbolizes the vehicles signature on a message m.*

2.2 Assumptions

Any secure system and its appropriate protocols are designed based on some assumptions. The goal of every secure system is to mitigate the threats and make it difficult for attackers to penetrate the system. It is well perceived that the assumptions of a VC system play an important role when it comes to security. All the functional and security requirements are defined based on the assumptions within the system and environment. Having considered assumptions in different stages of design and deployment, it can be said that threats are mitigates. Otherwise, attackers are able to violate system security policy using those assumptions. The current vehicular public key infrastructure (VPKI) has also been designed and deployed based on some assumptions. In the following, each of them is mentioned briefly.

Trusted Components
The assumption regarding trusted components (TCs) is that they are tamper-resistant devices and they operate without being compromised. In the case of physical attack, all the credentials are being discarded without being revealed. The roles of TCs are as follows:

- *To protect the vehicle's cryptographic material and their use* [7].
- *To safeguard data usable for liability identification* [7].

Limited Resources for Adversaries
The assumption is that adversary nodes have limited resources in terms of computational power and memory. Recollection of all messages by attackers is possible by summarizing them [7]; however, eavesdropping the whole network, including all the packets going back and forth, is infeasible. In other words, adversaries can collude to gather information or eavesdrop packets in order to stage a complex and gigantic attack; however, the assumption is that adversaries do not have global knowledge. They cannot enforce every node within the system to behave adversarial. The detection of these attacks, using intrusion detection mechanisms, and the recovery procedures shall be put in place to keep the VC system in a secure state.

No Omniscient Attacker
According to the assumptions, there exists no omniscient attacker with full knowledge of the vehicular communication network to perform different exploits. Each adversary is capable of gathering a partial knowledge of the network, although they can exchange information to collude and attack the system.

CA's Certificates Available for Every Vehicle

Having access to the CA's certificates, each vehicle is able to encrypt the messages with the corresponding CA's public key. In the case that CA's certificates are not stored on the vehciles' HSM, there must be a way to obtain it securely.[1]

No Resource Depletion, No Communication Jamming

An attacker can disrupt the traffic between Vehicles, RSUs, the PCA or an LTCA. This is a controversial issue and cares need to be considered. In this topology, the assumption is that the communication jamming never threatens the system or the communication protocols are designed in a way that they can detect and mitigate the communication jamming. A comprehensive discussion on jamming and resource depletion are provided at the end of this chapter, the adversary model, in section 2.6.

RCA

RCA (Root Certificate Authority) is the authority to issue certificates for LTCAs, PCAs and PRAs. Each RCA can be a national CA, or a government CA. The RCAs, on the top, can be cross-certified or certified by other mechanisms. The main focus of this report is on the relationship between ITS-Ss, PCAs, LTCA and PRA to achieve pseudonymous certificates, the up-to-date CRLs as well as resolving the real identity of a pseudonym. The relation between RCAs is not a matter of discussion in this report while different RCAs are sketched in the VPKI infrastructure, in figure 2.1. Further discussion is outside the extent of this study.

Valid Certificates are Propagated

As a boot-strap loading phase, the assumption is that valid certificates of RCA, PCA and LTCA are properly distributed and stored on the HSM of each vehicle.

Long-Term and Short-Term Certificates

In this architecture, long-term credentials are not used to secure the communication nor providing secrecy or anonymity. To provide anonymity for each vehicle, each node is equipped with multiple certified pseudonymous public keys, which can be acquired by a trusted third-party called Pseudonym Provider (PNP). Thus, pseudonym public keys and the corresponding private keys will be used for a short period of time, ε seconds, called the pseudonym lifetime and finally, it will be discarded [18]. Each vehicle has only one valid long-term certificate while it asks for pseudonymous certificates once every few days.

Resilience to Failures

The system is assumed to properly working and in the case of a fault or a failure, the system comes back to a secure state after a short period of time. In other words, it recovers itself from an insecure state to a secure state since the system always has to operate in a secure state.

Public Vehicles vs. Private Vehicles

Based on [19], there exist different types of vehicles such as: private vehicles, emergency vehicles, public transport vehicles and so forth. In general, public vehicles never concern about their privacy since no real identity is assigned to them. When it comes to talk about

[1] Vehicles might receive CA's certificates using out-of-band mechanisms.

private vehicles, the story is different. They are much concerned about their privacy and they do not want to be tracked or identified. As mentioned in [19], having different hierarchies[2] in the VPKI has some advantages. In the case of public vehicles, they are less concerned about their privacy while having different hierarchies makes it appropriate for administration and accountability issues. Private vehicles, on the other side, disseminate the vast majority of messages within such an infrastructure which might congest the network traffic or communication [19]. Vehicles are identified by their PSIDs, Provider Service Identifiers, to be differentiated as either private or public vehicles. In [19], public vehicles are referred as emergency vehicles while private vehicles are referred as basic vehicles.

In general, private vehicles need pseudonym certificates whereas public vehicles do not need pseudonymity. Thus, such differentiation depends on the design of the system. It can be the case that each public vehicle uses its long-term certificates to communicate. However, in the case that they might need some short-term keys, they can ask PCAs for short-term certificates. The PCAs issue short-term certificates for public vehicles, not necessarily pseudonym public-keys, but some traceable short-term keys. For private vehicles, cares need to be taken to the full extent to protect vehicles' privacy since protecting vehicles' and users' private information is one of the most important issues in a secure vehicular communication system.

Above all, there are some assumptions mentioned in details in [7], which have to be considered as well. For the sake of shortness, they are not explained in details here; however, the main operational assumptions regarding a secure VC system are as follows:

- *Authorities*
- *Vehicle Identification and Credentials*
- *Infrastructure Identification and Credentials*
- *User Identification and Credentials*
- *User and Vehicle Association*

In the following, functional and security requirements in the a secure VC system will be defined and explained. Afterwards, the infrastructure of VPKI and the communications among ITS-Ss, a PCA, an LTCA and a PRA are depicted. At the end of end of this chapter, an adversary model corresponding the secure vehicular communication system is explained. Precisely speaking, investigating an adversary model in designing a system helps to consider different threats within the system in order to put appropriate countermeasures in place.

2.3 Functional Requirements

The functional requirements within the VPKI are the most fundamental requirements which have to be considered and integrated in the design phase. These requirements are briefly explained below. It is crystal clear that not all of them are expected to be considered, implemented and deploy in this project. For the sake of completeness, all the functional requirements are listed. Some of them are denoted in chapter 6, the future works.

[2]Different hierarchies imply different vehicles and entities within the VC system.

Certificate Authorities

As described in details in chapter 1, section 1.4 definitions and key concepts, there are different benign authorities, considered as completely trustworthy organizations within the domain of VPKI. Their roles are to provide credential management in a secure vehicular communication system. Furthermore, they shall keep the system in a secure state, evict malicious entities and provide security services for legitimate nodes. On the top level, RCA is located, which is the trust anchor within the VPKI and its certificate is self-signed. PCA is the entity, issuing pseudonymous certificates for the vehicles. LTCA is another trusted entity within the domain of a secure VC system and its responsibility is to issue long-term certificates for the vehicles, RSUs and other entities. Finally, PRA is the corresponding authority to perform pseudonym resolution. It can query PCA and LTCA to resolve a pseudonym and in the case of need, revoke all the related pseudonyms. For further details, refer to chapter 1, the definitions and key concepts in section 1.4. Figure 1.1 in chapter 1 illustrates the relations between different hierarchies of authorities, including RCA, PCA, LTCA and PRA.

Public-Key Certificates

Each legitimate entity within the domain of a VC system shall be equipped with a certificate in order to perform security services. In the area of a secure VC system, each vehicle is provided with a long-term certificate, issued by an LTCA, and a finite set of short-term certificates, issued by a PCA. Since privacy is the most important requirement from users' perspective, pseudonym certificates are used to provide pseudonymity. Further discussion comes in the rest of this chapter, security requirements, in section 2.4.

IEEE 1609.2 mentions additional requirements to be provided by the VPKI [19]. The requirements are pointed out in terms of structure and role. As it turns out to the structure, certificates shall provide identity-based as well as role-based authentication [19]. The trusted enrollment authority shall be able to identify the real identity, based on the regulated policies [19]. On the other side, the roles in certificates aim at specifying some kind of permissions in order to manage individual identities, or a group membership [19]. For more information, refer to IEEE 1609.2 [19].

VPKI Hierarchy

When it comes to the relationship between different CAs and sub-CAs, it is well perceived that PKI relationship is a really huge research area. Briefly speaking, trusted root CAs have to be authenticated and certified to each other, based on a model and a policy. There are many schemes regarding top CAs trust models while each has its own pros and cons. As an example, if RCAs are cross-certified, each ITS-S shall be fitted with a single RCA certificate. In this case, it can verify the rest of the RCAs' certificates using the cross-certification schemes. However, if they are not cross-certified, each ITS-S shall be provided with all the RCAs' certificates in order to verify other RCAs' certificates [19]. As suggested in [19], the maximum certificate chain length is five including the levels of: EU, country, state, entity, sub-entity. Further discussion is beyond the reference of this study while [19] provides additional details.

Roaming

The idea of roaming in VPKI is similar to the cellular network that a user travels to other couturiers while his mobile phone is kept connected to the cellular network. In a VC system, each vehicle can cross between different realms while it needs to communicate,

send messages and receive messages from other foreigner entities. In other words, the VPKI has to support roaming to accept valid messages from foreigners; moreover, foreign vehicles shall be fed with required messages as long as they stick to the VC system policy throughout their session. To put it in another way, roaming is a contract among different realms in order to provide continuous connectivity and services to the legitimate entities [19]. Further discussion is irrelevant to this research study; [19] outlines more information and references regarding roaming.

VPKI Permission

IEEE 1609.2 v2 provides additional details about PKI permissions [20]. The idea of permissions in the domain of the VPKI is defined by IEEE 1609.2 using the concept of Provider Service Identifier (PSID) [20]. Having furnished each certificate with a list of PSIDs, the rights of certificate holders are determined [20]. In this way, VPKI has to consider *addition of permissions* as well as *removing permissions*. Detailed discussion on VPKI permission is beyond the scope of this research; however, a brief discussion would be interesting. The idea is that if the issuer's certificate gets compromised, how the issued certificates shall be treated. There are two main approaches proposed in [21] namely: *"shell-model"* and *"chain-model"*.

In the *"shell-model"* scheme, the validity of the issued certificates are directly affected by the validity of the issuer certificate. If the issuer's certificate gets compromised, all the issued certificates have to be immediately revoked. On the other hand, *"chain-model"*, which is less practical in compare with *"shell-model"*, states that the validity of the issuer's certificate does not affect the validity of the issued certificates. In the case of getting compromised, the issued certificates are still valid until the end of their expiry date [21]. For further discussion, refer to [21].

Update of RCAs' Certificates and Introduction of New RCAs

One of the challenging issues in the VPKI hierarchy is the introduction of a new RCA, or updating the certificates of the trust anchors. In the case that an RCA's private key is compromised, the security of the entire system is at stake. Mechanisms are required in order to manage updating RCAs' certificates as well as introducing a new RCA. Having introduced a new RCA's certificate, or updated an RCA's certificate, it shall be disseminated and stored securely among the legitimate entities within the VPKI [19]. Further discussion is outside the confines of this research project. In the future works, in section 6.1, it is explained more.

Obtaining Long-Term / Pseudonymous Certificates

The main focus of the project at hand is how to obtain pseudonym and long-term certificates in the domain of a VPKI.[3] An entire chapter[4] is dedicated to propose these protocols with great details. Further discussion, for the sake of shortness, is avoided.

Certificate Revocation

To keep the system always working in a secure state, the entities, reported as malicious or compromised nodes, shall be evicted from the system; moreover, the rest of the entities shall be informed about those revoked nodes. There are some mechanisms in order to

[3]To be more precise, to obtain long-term certificates, an open-source software which has implemented the standard protocol is used. The main focus is on designing and implementing short-term certificates.
[4]Chapter 3, *Protocol Design*, provides additional information.

deactivate the credentials and the certificates. The detailed discussion of how to figure out the malicious nodes is outside the sphere of reference while it is assumed that an appropriate intrusion detection system (IDS) is put in place. The dominant approach is to use CRL[5], Certificate Revocation List, to inform other entities about the revoked nodes. The challenging issue here is that since there exist two different certificates, long-term and short-term, there should be two kinds of certificate revocation lists. When it comes to revoke a pseudonym certificate, the real or the canonical identity of the pseudonym certificate holder is not determined; as a result, the corresponding trusted authority shall obtain the real identity in order to deactivate it [19]. In chapter 3, a protocol to obtain the latest short-term CRLs is discussed.[6] For further information, [6] provides comprehensive and detailed discussions on *"Certificate Revocation List Distribution in Vehicular Communication Systems"*.

Supporting Cryptographic Algorithms

To deploy and implement VPKI, some libraries will be used. The idea is to use the standard and open-source libraries, such as: openSSL [22], openCA [23] and Mozilla Firefox [24]. Those open-source libraries support many different cryptographic algorithms, such as: Elliptic Curve Cryptosystem, ECDSA, SHA-256 and so forth. The detailed comparison is discussed in chapter 4, the design and implementation.

Certificate Repository / Directory

Based on the standard PKI, there could be a directory in order to disseminate the certificates such as an LDAP directory. Although this is completely beyond the scope, it could be considered to distribute certificates, key backup and recovery purposes in the upcoming researches.

Key Management and Key Exchange

One of the main issues in the domain of a public-key infrastructure is key management and key exchange protocols. PKCS[7] provides an entire set of standards in order to perform security services. Further information is available online [25].

2.4 Security Requirements

There are some requirements to make such a broad system secure. However, the most controversial requirements, in the area of secure VC systems, are message authentication, integrity, non-repudiation as well as protecting users' private information, not allowing anyone violating their privacy. In the following, all the necessary requirements are explained while the main focus would be on the most controversial ones, as mentioned above.

The message confidentiality is not really the most-valuable issue since the goal is not to keeping a message unreadable from other entities; but there is a need to transfer a message correctly, intact as well as avoiding message repudiation. On the other hand, asymmetric cryptography provides the requirements to authentication, integrity and non-repudiation, as the primitive requirements. Moreover, users' privacy provisioning is not a fact to be

[5]There are other mechanisms to disseminate the revoked entities, such as: *"Non-renewal of expired credentials or tickets"* [19]. The dominant solution though is the distribution of CRLs.
[6]Obtaining long-term CRL is integrated within the system from the standard implementation of the protocol.
[7]Public-Key Cryptography Standards

provided by asymmetric cryptography. To achieve it, pseudonymity or pseudonymous authentication and privacy-enhancing technologies are used in order to protect privacy.

What needs to be clarified here is that in a normal communication to receive safety messages, environmental hazards, road conditions and infotainment, secrecy is not a matter of fact while integrity, authentication and non-repudiation are the dominant sides. However, it is really important to mention that in acquiring pseudonymous certificates, long-term certificates and CRLs, message confidentiality is also important which will be cleared up in the corresponding sections.

The requirements below have been gathered from [7] and [19]. To achieve such a secure system, these requirements have to be taken into consideration carefully.

2.4.1 Message Authentication and Integrity

Message authentication and integrity are the most important factors to prevent message modification and forgery [5]. Each and every message has to be traversed intact so that the receiver can process and perform the required actions. In other words, these requirements protect messages from alteration, modification and fabrication. Furthermore, the receiver must confirm the identity of sender [7]. To mitigate the threats on data in transition and also, to keep the data consistent, there exist mechanisms which will be used to protect message authentication and integrity in the communications within the VC system.

2.4.2 Message Non-Repudiation

This service is used to prevent entities from denying sending, receiving or transmitting a message [5]. One of the mechanisms to provide non-repudiation service is using asymmetric key cryptography, which will be used for the communications in the protocols within the VC system. Having signed a message using the private key, the signer cannot deny sending the message since the private key has to be protected secretly by its owner.

2.4.3 Privacy

Privacy protection, as the most important requirement from the users' standpoint, is applying mechanisms to prevent collection, extraction or inference private information from vehicles during the communications [5]. The policy should protect personal information to the full extent and it should never reveal private information of users to illegitimate authorities [7]. Privacy shall be considered from the very beginning of the system design using privacy enhancing technologies and mechanisms. In other words, the idea is to protect privacy not only from legal point of view, but also from technical aspects in terms of using privacy enhancing technologies. Although privacy protection is considered a broad area as a requirement, there exists another narrower requirement called anonymity.

2.4.3.1 Anonymity

Taking into consideration, the difference between confidentiality and anonymity is that confidentiality is keeping the content of the messages, going back and forth, secret from unauthorized entities; while anonymity is keeping the doer of the action secret or making it difficult (or impossible, in the case of full-anonymity) to identify the identity of the node who did the transaction. In other words, in anonymity the identity of the entities who

committed the transaction is not completely exposed, while legitimacy and authenticity of them shall be validated. As defined in [25],

> "Anonymity is the state of being not identifiable within a set of subjects, the anonymity set."

In a VC system, the goal is to allow authenticated entities to commit a transaction without being identified with their real identities and the system guarantees unlinkability.[8]

There are mechanisms to provide anonymity within a VC system so that an observer cannot make a link between messages to identify the real identity of a node since the probability of performing an action is identical for each node [7]. However, in the case of law, a legal authority is perfectly capable of figuring out the real identity by making links from different nodes and entities. In other words, some benign entities, completely trustworthy organizations, are able to link the transactions in the case of forensics investigations. Except the legal authorities, the protected information shall be preserved from unauthorized entities so that they cannot be exposed through the execution of any sequence of commands [7].

As alluded in [7], anonymity is not considered as a mandatory requirement for all the entities within the vehicular communication system. Based on the assumptions mentioned above, there exist two kinds of vehicles called private vehicles as well as public vehicles like emergency vehicles or public transportation. Private vehicles are completely concerned about their privacy while being anonymous in performing all the activities. Moreover, it is really important for them not to be monitored or tracked. On the other hand, a subset of the infrastructure nodes are considered more trustworthy [7]; thus, public vehicles are not concerned in terms of privacy owing to the fact that public vehicles are not assigned to a unique identity. As a result of this, it can be inferred that providing privacy and anonymity for users within the VC system can affect the system satisfaction intuitively.

In ETSI likewise, privacy protection has been categorized from different viewpoints in terms of protection against violation either by (valid or invalid) third-party ITS-Ss or by authorities (CAs) [19]. As suggested in the standard, privacy violation against authorities can be thwarted in organizational level, by splitting the roles among different CAs (separation of roles). On the other side, privacy violation against third-party ITS-Ss shall be mitigated from technical aspects, by using pseudonymous credentials on a regular basis, and changing them every few days [19]. Both mechanisms have been applied within the VC system to protect privacy.

Should a new pseudonymous credential is used for a short period of time, the IP and the MAC[9] address of the corresponding vehicle have to be changed as well. The discussion on how to achieve it in lower layers is completely outside the sphere of reference of this research. However, the reason behind it would be of great interest. The philosophy behind is that if an eavesdropper can track and monitor vehicles using their IP and MAC addresses, then it would completely violate the vehicles privacy, even though pseudonym certificates are being used perfectly. Anyway, the state-of-the-art approach guarantees

[8]The difference between anonymity and pseudonymity is that anonymity guarantees that the transaction is kept completely anonymous while pseudonymity does not guarantee. In other words, using pseudonymity, an attacker can make a link between different pseudonyms after a period of time. Moreover, providing anonymity costs more than providing pseudonymity from computational and communication point of view.
[9]Media Access Control

that the IP address as well as the MAC address on the lower layers[10] will be changed if the pseudonym credential is changed.

2.4.3.2 Unlinkability and Unobservability

Unlinkability and unobservability are two important privacy-related concepts, considered necessary when it comes to equip a system with anonymity, unobservability, or pseudonymity properties [25]. Unlinkability states that a threat agent[11] is not able to make a link between two or more items of interest (IOI) within a system to identify the real identity [25]. In other words, having equipped a system with unlinkability properties, a user can perform different transactions and use multiple services without being identified [25].

On the other hand, unobservability implies that a user can use multiple services and resources within the system while no one can figure out these services or resources are being used [25]. To put it in another way, unlinkability deals with making a link to identify the real identity while unobservability is to keep the operations, being used by users or entities, secret [25].[12] It goes without saying that identification is required to allow legitimate entities communicate within the domain; however, observing a specific entity or a vehicle to identify its identity, or to figure out the operations, transaction or services, which are being used, shall be infeasible in order to provide anonymity.

2.4.4 Entity Authentication

This requirement is dealing with the aliveness of the sender and as a result, the message [7]. This ensures the receiver to provide evidence about the aliveness of the sender as well as the messages. Message authentication and integrity ensures that the message has been traversed and received intact while entity authentication proves that the message is generated from an alive sender within an approved interval [7].

2.4.5 Message Confidentiality

Keeping the content of the messages secret from unauthorized entities is the main factor in message confidentiality [7]. Only authorized entities who have access to the keys are able to interpret messages.

2.4.6 Access Control

Access control policies determine which services can be used by which entities. Moreover, they specify what each node is allowed to do and not allowed to do [7]. This service includes identification, authentication as well as authorization.

2.4.7 Accountability

Capability of being audited by legitimate authorities is considered a crucial operation within the system. All messages and protocol executions are logged and recorded so that forensics investigations can be done in the case of malfunctions or illegal operations [7].

[10] Network layer and link layer are the responsible layers to perform it.
[11] An attacker or an adversary
[12] Unlinkability is more user-focused while unobservability is less user-focused and is more general to the operations and used services [25].

2.4.8 Availability, Fault-Tolerant and Robustness

The design at hand seeks for an approach so that the system remains operational and the services are available in the case of faults. Moreover, the system is fault-tolerant in the presence of malicious or benign failures and also, it can be robust and resilience to resource depletion attacks [7]. Generally speaking, designing such a fault-tolerant and robust system needs a lot of efforts to do and many issues to consider; this requirement is considered as one of the most important pillars of security.

2.4.9 Liability Identification and Forensics Investigation

Providing information to identify or assist the attribution of liability may stem from the current practice in transportation systems [7]. The current practice in nowadays transportation systems makes all drivers liable to the activities they do. On the other side, legitimate entities are able to investigate all deliberate or accidental actions in the case of law [7]. As a result, a legitimate authority can extract all the information required to investigate an accident in order to figure out the chain of custody.

2.4.9.1 Pseudonym Resolution

As brought up earlier, privacy is the most important requirement in the domain of a secure VC system. As a result, no single entity should be able to make a link and figure out the real identity of a pseudonym. In other words, protecting users' privacy implies that tracking and monitoring users to figure out the real identities should not be possible.[13] The roles of different authorities are separated so that not a single entity has access to the entire information. The PCA has partial knowledge while the LTCA has the rest of the information. If they collude to share their information, then the efforts of separation of role would be pointless. However, under specific circumstances[14] for law enforcement, the real identity of a specific pseudonym in a transaction has to be identified[15]. This has to be performed by a legitimate authority using access to different CAs (PCA and LTCA). This legitimate authority is called PRA, Pseudonym Resolution Authority. Thus, making a link to figure out the real identity of a pseudonym is called pseudonym resolution. This does not affect or violate users privacy since pseudonym resolution is done in the case of malicious behavior or hazards under the supervision of the a trusted authority. The details of pseudonym resolution is explained in section 3.5 with great details.

2.4.10 Scalability and Performance

One of the problems in all the centralize-oriented approach is the lack of scalability. Based on the resources available, the dominant approach is selected to be centralized. However, it lacks the scalability feature. In the case of a failure in a server, maybe in one of the PCA, LTCA, PRA, or even RCA server, the system encounters a terrible and serious problem since redundancy and replicability are not supported within the system. The assumption here is to have a system, which remains operational, even in the case of failures in each of the RCA, PCA, LTCA or PRA. The recovery mechanisms are totally

[13] It is impossible in a reasonable period of time using limited resources.

[14] The secure VC system policy determines who, when, why and under which circumstances a trusted authority can perform a pseudonym resolution. Such circumstances could be an accident or misbehavior of a vehicle.

[15] Due to the non-repudiation requirement, if a vehicle sings a message, it cannot repudiate it afterwards.

beyond the reference of this research study. However, keeping it in mind helps to enlarge the system in the long run.

Based on the security requirements, the corresponding security mechanisms have to be put in place to mitigate different abuses. Such security mechanisms are pointed out in chapter 3, when it comes to explain the details of the protocols. To safeguard the system operations, appropriate security mechanisms and countermeasures shall be considered in the design phase.

2.5 VPKI Scheme

Figure 2.1 shows the infrastructure of the vehicular communication system. The main entities are RCAs, PCAs, LTCAs, PRAs, ITS-Ss, Vehicles and RSUs. The figure clarifies all the communications between a PCA, an LTCA, a PRA and ITS-Ss.

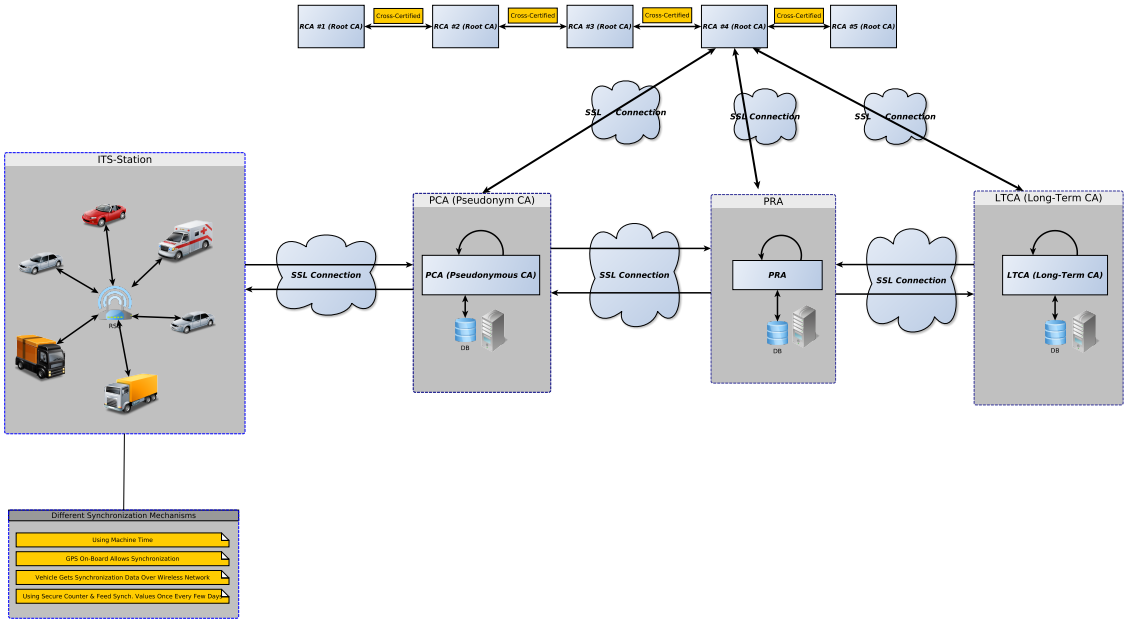

Figure 2.1: VPKI Scheme

As elaborated in figure 2.1, RCAs are either a hierarchical structure or a cross-certification structure. For the research at hand, these relationships will not be discussed and for the time being, cross-certified approach is selected and shown.

Within VPKI, there exist other entities such as vehicles and RSUs, called as ITS-Ss. Each vehicle has a unique and identifiable identity, V, and a pair of public key and private key, K_V and k_V, respectively, registered with exactly one CA [6]. Evicting nodes from the system and propagating the CRLs are the responsibility of the CAs[16] to safeguard the system operations. In other words, they do not allow any small subset of malicious nodes to perform adversarial activities or maliciously accuse and evict legitimate nodes [6].

[16]It means all the CAs including RCAs, PCAs and LTCAs are the responsible authorities to disseminate CRLs. PCAs are the responsible entities for pseudonym CRL while LTCAs issue long-term CRL.

The assumption here is that the VC policy does not allow vehicles to be registered within the domain of different CAs. In other words, each vehicle can receive only one long-term certificate from an LTCA. Since the LTCA is involved in the process of issuing pseudonymous certificates, a vehicle can send request to achieve pseudonymous credentials to different PCAs. This implies that a vehicle can receive pseudonymous certificates from different PCAs. To make it more clear, each vehicle is registered with only one LTCA while it can acquire pseudonymous certificates from different PCAs.

In figure 2.1, different vehicles including private vehicles[17], public vehicles and emergency vehicles are sketched. Based on the standard [19], privacy is not a mandatory requirement for public vehicles[18] since they are not assigned to specific persons. Thus, public vehicles never request to achieve pseudonym certificates. However, they are considered as a trusted portion of the secure VC system. They might be the nodes for disseminating new information or they might be utilized for management goals. Each of them has a long-term certificate while they apply also to acquire CRLs from LTCA and PCA. To put it in nutshell, privacy is a mandatory security requirement only for private vehicles.

2.6 Adversary Model

There are many attacks and exploits which could jeopardize the system operation and its performance from security point of view; a variety of security breaches in different systems have illuminated that every system is vulnerable to different exposures. There exist many exposure in different levels and parts of the systems which have to be considered to thwart threats. Hence, the goal of every adversary model is to mitigate the threats and make it difficult, in terms of time and cost, to attack a secure system.

A general adversary model for the vehicular communication system is explained in [7] to identify the attacks which could affect the VC system from security perspective. Each system is vulnerable to different abuses, threats and attacks and the VC system is not an exception, indeed. The problem at hand is how to keep the VC system operating always in a secure state. In other words, the aim is to secure the operations of the VC system to the full extent so that it mitigates all possible attacks and thwarts the threats. It is targeted not to allow an imposter to stage an attack using the vulnerabilities and weaknesses within the system [7]. The protocols for secure VC systems are supposed to be designed in a way that an adversary node cannot deviate from the protocol definitions; subsequently, staging an attack and jeopardizing the VC system from the security aspects get difficult.

Based on the general adversary model in [7], network entities have been categorized into correct or benign and faulty or adversaries. However, the main focus is on the behavior of the adversarial nodes to handle illegitimate and destructive activities. Dealing with benign failures is not a matter of focus, though; some of them have been mentioned in [7], such as: crash failures, omission failures, and timing failures.

The adversaries, who can threaten and deteriorate the VC system security, have been classified into active and passive adversaries [7]. The adversary model in [7] grouped illegitimate and destructive activities within a VC system into 5 categories, as follows:

- *Localized and Selective Denial of Communication*

[17] Private vehicles are known as personal vehicles.
[18] Public vehicles consists of buses, emergency vehicles, trucks, taxis, police cars and the like.

- *Internal Active Adversaries*
- *Bounded Adversarial Presence*
- *Input-Controlling Adversary*
- *Other Adversary Models*

For the sake of shortness, the detailed definition of the above outlines will not be discussed explicitly. For further information, refer to [7]. Here, the possible security threats in designing protocols to acquire pseudonymous certificates and short-term CRLs are addressed. The idea is to map the existing threats into the proposed adversary model, based on the assumptions and the requirements, to mitigate the threats and utilize appropriate countermeasures.

Referring to figure 2.1, the threats within the VC system, to acquire pseudonymous certificates, short-term CRL acquisition as well as pseudonym resolution will be elaborated below.

2.6.1 Localized and Selective Denial of Communication

In a VC system, it is possible for a bad guy to jam the communication and disrupt the communication among entities in a way that they cannot communicate with each other properly. It is enough to discard the requests from an ITS-S towards a PCA or an LTCA. Thus, inability to receive pseudonymous certificates as well as the latest CRLs are attributed to the communication jam, staged by an adversary.

Assume that each vehicle is scheduled to receive pseudonymous certificates on Saturday morning between 8-10 every week. If an intruder jams the communication during that period, the vehicle cannot receive pseudonymous certificates for the next week and as a result, it is exposed to a variety of threats such as privacy infringement. The attacker managed to violate system security policy and the VC system is not properly protected. It is applicable also to receive the latest CRLs. Inability to communicate and receive pseudonymous certificates or CRLs is giving rise to breach system security policy. Taking into consideration that preventing a system from denial of communication is outside the extent of this essay and the assumption is that resource depletion and communication jam never happen, or in case of happening, it can be automatically recovered using a mechanism.

2.6.2 Internal Active Adversaries

There exist different internal active adversaries like: modification, forgery, replaying and colluding to infer information and stage a possibly enormous attack. In the following, each of them is considered to be clarified.

Modification and Tampering

The idea of modification is to modify messages in-transit to be either meaning-full or meaning-less. Message modification is possible within the system for the current protocols since they are sent via wireless (IEEE 802.11p). However, the integrity of messages are protected using message authentication code (MAC) mechanism. To alter a MAC, one has to know the corresponding key. Furthermore, the MAC of a message is signed by the sender, not to allow an eavesdropper to modify and re-transmit a faked message. In case of modification, the recipient will recognize and notify the sender about the corruption.

Thus, message modification cannot threaten the system security as a whole since it can be easily detected. However, an adversary is capable of corrupting all the messages for ever and subsequently, a resource depletion attack is staged. The sender cannot figure out that an attacker is discarding the messages since there is a chance that the recipient is not available or a technical problem has been arisen. The assumption is that an attacker cannot control the whole network. This one is categorized in localized and selective denial of communication section, which has been covered earlier.

Forgery

Ability to forge a message or impersonate an identity require to compromise a node or infect an entity. Keeping the credentials and cryptographic keys secret from unauthorized entities is really critical while it cannot guarantee the correctness of operations in a system [7]. Credentials and cryptographic secret keys can be compromised and this might result in forgery and masquerading. Thus, impersonating an identity and injecting bogus messages is possible in the case of credential compromising; the assumption is that correctly implemented cryptographic primitives are considered secure [7]. In brief, message forgery does not threaten the security of the system since integrity service has been considered in advance.

Recollecting Past Messages

Another active attack is that an adversary can recollect past messages to infer information and perform an attack. The idea here is to choose which message(s) (or which part of a sequence of messages) has to be forged, replayed, modified, omitted or delayed [7]. Modification, fabrication, and interruption of messages can be either message content dependent or independent [7]. For the three protocols, proposed in this report, an appropriate countermeasure is put in place.

To acquire pseudonymous certificates, the latest CRLs or performing pseudonym resolution, a requester has to establish a new session to send its request. Having received the reply from the corresponding recipient, the session will be expired and it is no longer valid. Thus, recollecting messages does not profitable for an eavesdropper to infer more information from prior sessions as long as the communication credentials are correctly changed and properly protected. An attacker can conclude from different sessions if they are somehow correlated with each other, or the same key is used for different sessions. As a result, by recollecting all messages, an adversary cannot interpret them to stage an attack since messages are cryptographically protected in a proper manner.

Replay attack, re-transmitting an old message to fool the recipient, is also protected by using nonce and time-stamp. The assumption is that ITS-Ss, including vehicles and RSUs, as well as other authorities (RCAs, PCAs, LTCAs, PRAs) are time synchronous using any mechanism.

Multiple Adversarial Nodes

Until now, the assumption was that each attack is staged by only one attacker, since attackers exploit independently. However, there is a chance for attackers to collude and perform a complex attack. They can coordinate and exchange information to do a serious attack and harm the system. Due to the complexity of such attacks, thwarting the threat needs also a lot of efforts to do and many countermeasures to apply in advance.

This is a really applicable attack for the protocols which will be proposed next. A possible scenario regarding multiple adversarial nodes for these protocols is that there

exist two trusted third-parties called PCA and LTCA, which work separately, and their roles are individually assigned in advance. However, the roles between these two entities have been divided so that each entity has not the whole knowledge of the system, but a partial of it.

As a scenario, suppose that PCA and LTCA collude to identify the real identity of a vehicle after it requests pseudonymous certificates. Hence, colluding and exchanging information lead to information disclosure and privacy infringement. Thereby, vehicles are exposed to be tracked and monitored even though they use pseudonymous certificates to protect their privacy. It is crystal clear that two trusted entities within the VC system never collude to violate system policy, based on the assumption of trusted third-parties. However, having access to an LTCA database, there is a possibility for an eavesdropper to sniff the communication between ITS-Ss and a PCA to infer information. Later on, an adversary can correlate data to stage an attack or identify the identity of a vehicle. Figure 2.1 draws the attention to the secure communication between the PCA and the LTCA. The pre-established channels are not vulnerable to eavesdropping since the communications among entities are encrypted and integrity-protected. As a result, each vehicle has to establish a secure channel before starting the communication. In chapter 3, the protocol design, it will be more clear how each vehicle uses SSL[19] to make the communication secure.

Another scenario is that several vehicles with adversarial intentions agree to collude on disseminating bogus information. Such malicious vehicles cannot propagate messages without signing. It is obvious that in the case of accident or undesirable events as a consequence of such bogus messages, legal authorities can investigate the scenario to catch the guilty nodes. Therefore, colluding and coordinating to perform an illegal activity can be monitored and traced back to identify the attacker.[20]

Performing an adversarial activity without being identified is indeed possible by compromising the credentials and cryptographic keys. However, the assumption here is that correctly implemented cryptographic primitives are considered secure [7] and in the case of compromise, the compromised nodes are quickly evicted and reported to be excluded from the system and added into the CRL. The protocols, proposed in chapter 3, are designed in a way not to allow nodes to perform an activity anonymously. This means that they have to be a legitimate entity, registered within the system; furthermore, every transaction is being logged for forensics investigations. This is not in contrast to privacy protection since tracking and tracing back the transactions are possible by legitimate authorities and only in the court of law.

An attacker is also capable of changing the number of compromised nodes within the domain of a VC system. This is called *"adaptive adversary"*[7], as another kind of internal active adversaries. Having subverted a node, an attacker can select which exact nodes remain compromised during the execution of the protocol [7]. Moreover, an adversary might subvert a legitimate node within the system and control that node for a period of time. This is called *"mobile adversary"*. Based on the assumption definitions and requirements, these two types of adversary are considered irrelevant in this context.

The final point is that when it comes to request for a pseudonym certificate or the latest CRLs, each vehicle uses the PCA's IP address, or the LTCA's IP address. It implies that

[19] Secure Socket Layer

[20] The idea is that each vehicle has to be legitimately registered within the domain of a VC system. If not, it cannot propagate messages since the rest of the vehicles never trust them. However, in the case that a legitimate node disseminate a bogus message, it can be traced back.

each vehicle sends its request directly to the recipient using its IP address. As a result, no routing happens among vehicles to reach the destination; thus, vehicles do not involve in routing. In other words, the threat of dropping packets or discarding messages by malicious vehicles on the way does not exist.

2.6.3 Bounded Adversarial Presence

According to [7], there has to be some boundaries regarding adversarial presence. For example, the number of compromised nodes, either in public vehicles or private vehicles, have to be bounded. In other words, not the majority of the vehicles and nodes within the system can act as adversarial nodes. There is a chance for a vehicle to be surrounded by adversarial nodes; however, it is rejected to have very few benign vehicles while the rest is malicious nodes [7].

2.6.4 Input-Controlling Adversary

Threatening a system does not always require to access the credentials and cryptographic materials, or compromise the cryptographic keys [7]. Violating a system security policy is also possible by altering the local inputs into the protocol [7]. This might be the case since deviating from the implemented protocols is difficult or time consuming; while it is considered sometimes easier to violate the system policy by altering the inputs or feeding the device with false data in compare with breaking the protocols.

Since an imposter is not able to perform any arbitrary behavior, he provides some *"malicious"* input values in order to induce the nodes to behave malfunction and accordingly, breach the system security policy [7]. In other words, the adversary does not deal with tampering the devices or altering the messages in-transit; rather, he tries to alter the input values from the origin. The on-board processing units are designed and deployed to be tamper-proof; this means that no entity can extract information from the unit.[21] However, the validity of the input data is not verified by the OBU[22]. To put it in another way, the tamper-proof devices such as OBU can be fed with false input while they cannot verify the correctness of the input data. Such input data can be originated from sensors or a GPS. It is assumed that the clock is an internal device and it always generates valid data. An attacker cannot inject bogus time-stamp since the clock is a part of the HSM, hardware secure module. Since GPS is located outside the HSM, there is no control over it and as a result, an intruder can inject bogus location information. The OBU signs the input data using the private key without validating them.

Provided that the inputs are originated from a secure module, an attacker cannot influence on the correctness of the inputs and as a result, *"input-controlling adversary"* is thwarted. However, the problematic issue is that not every sensor is located within the HSM. For instance, the clock is located inside the HSM whereas the GPS is not. Thereby, the inputs generated by the clock are trustworthy while the location information might be bogus. There exist other sensors, located outside of the HSM and subsequently, they might be tampered with. In the protocols, proposed in chapter 3, the assumption is that the inputs are trustworthy and faked data are not fed to the devices. Further discussion about it is out of scope; however, for the sake of assurance, formal proof is necessary.

[21] The private keys, both the long-term private key and pseudonym private keys, are stored inside the HSM, Hardware Security Module.

[22] On-Board Unit

When it comes to acquiring short-term certificates, bogus data cannot be injected into the system to affect in the correctness of the protocol. The time-stamp is located inside the HSM while the rest of the sensors, located outside, do not really affect the correctness of the protocol. As a result, an attacker cannot produce bogus input or inject any invalid data to mislead the corresponding authorities. However, formal verification on the protocols is a must.

To achieve the latest CRL, an attacker cannot affect the system functionality by controlling the input or injecting false data. The detailed discussion is completely beyond the scope of this research study. However, a brief explanation would be interesting. When it comes to retrieve a CRL from an authority, it is totally straightforward since an attacker cannot inject bogus data in the process.[23] However, when it comes to achieve the latest CRLs using a V2V communication, there is a chance to mislead vehicles by false data. Having colluded, adversary vehicles can mislead a legitimate and benevolent vehicle about a CRL; they pretend to have the latest version of the CRL while they avoid sending, or they send an older version. Since that vehicle has no access to the PCA to verify, it accepts it. In this case, there is a need to reach a consensus about the validity of the data or the validity of the revoked nodes and the CRL. Other mechanisms would be voting or reputation scheme.

2.6.5 Discussion of Other Adversary Models

As alluded in [7], there are two dimensions towards the discussion of adversary model. The first is *Byzantine* adversary, dealing with nodes registered legitimately within the system while acting as adversarial. The second one is having illegitimate nodes who can eavesdrop any message while they can intercept and inject messages; such adversary nodes are not legitimately registered as a part of the system. They are called *Dolev-Yao* (*DY*) adversaries [7]. The assumption is that an intrusion detection system does exist to detect such malicious behavior. In other words, the system shall be designed in a way so that legitimate nodes cannot perform illegitimate activities anonymously without being identified. Moreover, illegitimate entities and malicious behaviors are detected and responded. This is a really broad area and is outside the sphere of reference of this research project.

To achieve pseudonymous certificates, *byzantine* behavior is not applied practically since each vehicle has to request individually and directly to the PCA and the LTCA to acquire pseudonymous certificates. Thus, the nodes on the way cannot relay the messages and behave as *byzantine* nodes; they can eavesdrop and receive messages or jam the communication while they cannot interpret the them.

However, when it comes to CRL dissemination, the story is different. CRLs can be propagated not only via vehicle-to-infrastructure (V2I) communication, but also using vehicle-to-vehicle (V2V) communication as a more advanced approach. V2V approach is used specifically when the RSUs are out of reach. Vehicles communicate with each other to retrieve the latest CRLs without congesting the network or occupying the infrastructure network capacity. This is like a peer-to-peer (P2P) approach, considered as a scalable approach. In the case of V2V, if many ungenerous vehicles (not really adversarial) surround a vehicle and they never give the victim the latest CRLs, or propagate the older versions of the CRLs, the victim never thinks of adversarial behavior. Instead, the victim thinks that its neighbors also do not have the latest and up-to-date CRLs. Consequently,

[23] The detailed discussion on how to achieve latest CRLs from a PCA will be provided in chapter 3.

if the victim cannot connect to the infrastructure or find a benevolent vehicle, it is never able to update its CRLs.

DY adversary states that any message transmitted within the system is capable of being eavesdropped by an active eavesdropper. Furthermore, it is possible to receive and delete any message or initiate a new conversation with any other nodes within the system to stage a Man-in-the-Middle (MitM) attack [7]. According to the assumptions, a *DY* adversary must have access to cryptographic keys, which requires an entity to be legitimately registered within the public key crypto-system in use, or compromise a node to gain its credentials. Otherwise, it cannot interpret the message contents as well as its digest without acquiring appropriate keys [7].

What needs to be emphasized more is the assumptions, which are agreed on; adversaries are bounded in terms of resources and time. Moreover, the probability to forge a signature or invert a one-way function to re-produce a faked message using its MAC or hash value is very low [7].

The gap between compromising a node, on the one hand, and detection, declaration, appending to the CRL, disseminating CRLs and retrieving them, on the other hand, is an important issue. The delay shall be as short as possible. The detailed discussion and evaluation of this task is beyond the reference of this study. This reminds the typical *zero-day attack*, which has to be considered as well.

Both PCA and LTCA have to consider some limitations regarding vehicles' requests. If legitimate vehicles behave in a byzantine manner, such authorities might be unreachable for other legitimate vehicles. Vehicles might start bombarding the authorities to achieve either short-term certificates or long-term certificates, or obtaining CRLs. To prevent such kind of byzantine behaviors, the corresponding countermeasures have to be put in place.

There are some other items that have to be considered while they were not mentioned explicitly in the model such as: benign faults, installing a rogue version of the VC protocol stack, or installing faked devices. Although benign faults, installing faked devices to mislead nodes, or installing a rogue version of a VC protocol stack are important, they are not a matter of discussion in the adversary model. The main focus of the adversary model is on intentionally attacking a system to change the system status from a secure state to an insecure. Further discussion regarding such issues is outside the confines of this thesis project.

In the next chapter, the details of the three protocols to obtain pseudonym certificates, pseudonym CRLs as well as performing pseudonym resolution are addressed.

Chapter 3

VPKI: Protocol Design and Implementation

3.1 Introduction

Every secure system requires a concrete design phase in order to be resilient to failures, and to be fault-tolerant and robust against attacks. Software defects in design-level, identified as flaws, are considered 50% of the whole errors and defects within a software system [26]. Thus, cares need to be taken when it comes to design, and more specifically, designing such a critical VPKI system. To achieve a flawless design, there is a need for some kind of formal verification techniques to prove the correctness of the protocols.

When it comes to cryptosystem, there are many mechanisms to apply within such a VC system. Due to the limitations, in terms of processing power and bandwidth, RSA and DSA signatures are not used [3]. To achieve the same level of security though, Elliptic Curve Cryptography (ECC) is selected since it provides the same level of security while it has less overhead and high-speed in compare to RSA and DSA algorithms [3]. As a result of this, each pseudonym certificate and private key is smaller than the similar keys in RSA and DSA. Moreover, performing a signature on a request using EC algorithms is much faster. Owing to the mentioned properties, IEEE 1609.2 has proposed EC-DSA signature to apply in the VC system [3].

Exploring further on the details of cryptogrpahic algorithms is beyond the reference of this study. However, a brief explanation is worth. The Elliptic Curve Digital Signature Algorithm (ECDSA) is used in VANETS for many different reasons. As displayed in table 3.1, *"the strength-per-key-bit is substantially greater in an algorithm that uses elliptic curves"* [27]. Using ECDSA for VANETs is due to the smaller size of ECDSA keys; this results in *"faster computation time and reduction in processing power, storage space and bandwidth"* [27]. Thus, it is ideal to be used in VANETs with all the limitations in terms of network bandwidth and storage space.

As indicated in table 3.1, ECC 160-bit key length provides the same level of security as RSA 1024-bit key length. Moreover, ECC 256-bit key is identical to RSA 3072-bit

key in RSA in terms of security level. For further information on a detailed comparison between ECC and RSA/DSA, refer to [27].

Table 3.1: *Security Level Comparison of Symmetric, RSA and ECC (key sizes in bits) [27]*

Symmetric	**RSA/DSA/DH**	**ECC**	*Time to break in MIPS years*
80	1024	160	10^{12}
112	2048	224	10^{24}
128	3072	256	10^{28}
192	7680	384	10^{47}
256	15360	512	10^{66}

ECC has several significant strengths over other type of cryptosystems while no weak point is reported so far [27]. As stated in [27], some of the them are as follows:

- *"It provides greater security for a given key size"* [27].
- *"It provides effective and compact implementations for cryptographic operations requiring smaller chips"* [27].
- *"Due to smaller chips, less heat generation and less power consumption"* [27].
- *"It is mostly suitable for machines having low bandwidth, low computing power, less memory"* [27].
- *"It has easier hardware implementations"* [27].

In this chapter, three protocols are described carefully with great details. These are the protocols for the communication among ITS-Ss, PCAs, LTCAs and PRAs. The first protocol is the communication between an ITS-S, on one side, and the LTCA/PCA, on the other side, to obtain pseudonymous certificates. The second one is dealing with receiving the latest short-term CRLs. Next, the communication protocol between an authority such as police, one one hand, and the PRA, PCA and LTCA, one the other hand, is expressed in order to perform pseudonym resolution. It goes without saying that the protocols are designed only for vehicles to achieve pseudonymous certificates and the CRLs. The trusted authorities can receive long-term certificates, or the long-term CRLs using out-of-band mechanisms while they do not need pseudonym certificate or pseudonym CRL. Before digging into the details of the protocols, a brief overview of SSL/TLS[1], regarding the current issues, is indeed necessary to be more clarified.

3.2 Protocol Scheme Outline

The bootstrapping phase is to establish a reliable, encrypted and integrity-protected channel in order to mutually authenticate two parties and communicate securely. To achieve this goal, transport layer security standard called TLS[2] is used. Based on TLS standard[3], two parties first mutually authenticate themselves and subsequently, the rest

[1] Secure Sockets Layer/Transport Layer Security

[2] http://tools.ietf.org/html/rfc5246

[3] TLS is the IETF standard version of SSL while SSL is the proprietary implementation by Netscape. IETF released TLS for the sake of compatibility in industry. In this report, SSL and TLS are being used interchangeably. For more information on SSL, refer to: http://tools.ietf.org/html/rfc6101.

of the communication is encrypted and integrity protected [28]. Since in the domain of VPKI, each vehicle is equipped with a certificate, the mutual authentication phase can be performed. The details of SSL, how it works and the like are not a matter of discussion here. For further reading, [28] provides an entire chapter on SSL in a simple and descriptive manner. However, concisely speaking regarding the VPKI and the upcoming protocols, having established an SSL channel, a master secret is used to generate 6 secrets, which will be used for integrity keys, encryption keys and IVs, 3 keys for each one. To provide integrity, it uses HMAC using the Integrity Key, IK. In the following protocols, if a vehicle wants to send data to the PCA, it uses its *write* key to encrypt while the PCA uses its *read* key to decrypt the data. However, in the protocol descriptions below, only one key is identified as the encryption key between two parties, which is a Session Key, *SK*.

To be more clear about the notations in all the subsequent protocols, the signature of authorities and vehicles are represented below:

- $\sigma_{LTCA}(m)$*: illustrates the LTCA signature on a message m.*
- $\sigma_{PCA}(m)$*: specifies the PCA signature on a message m.*
- $Cert_{PCA}(K_V^i)$*: describes the PCA signature on a short-term public key* K_V^i*. It symbolizes a pseudonym certificate.*
- $\sigma_V(m)$*: determines the vehicles signature on a message m using its long-term certificate.*
- $\sigma_V^{K_v^i}(m)$*: shows the vehicles signature on a message m using its short-term certificate.*

3.3 How to Request Pseudonym Certificates

To commence the protocol, both sides of the communication need to be authenticated to each other and perform a mutual authentication phase. Normally, establishing a secure channel is done in two different phases: Authentication and Key Agreement (AKA), Secure Communication.

Establishing a secure channel to communicate might be an expensive phase since every single vehicle is moving around a certain region which is not static.[4] The assumption is that vehicles are not out of reach and they can always establish a secure channel. Based on the standard, to set up a secure channel, each vehicle shall perform two phases. The first step is to carry out authentication and key agreement (AKA) phase to mutually authenticate each other and share secure session keys for the rest of the communication. Having the session keys shared, the bootstrapping phase is done. In the second phase, the secure communication starts and each vehicle sends its request to the corresponding authority. In other words, the AKA phase is the same for the following 3 protocols; however, in the second phase, each vehicle sends different requests to achieve its goal.

The procedure of acquiring new pseudonymous certificate in VPKI is described in details here. The assumption is that PCAs and LTCAs are always on-line and the long-term certificates of ITS-Ss have been already distributed and securely stored. There are two ways to design the protocol, the online and offline approaches. The online approach expresses that PCA shall queries LTCA to verify the token while in the offline

[4] A good idea to enhance the performance is to carry out all the required activities on the spot when a secure channel is established. For instance, a vehicle can send its request to the PCA to achieve pseudonymous certificates as well as the latest CRLs when the secure connection is set up.

approach, PCA is able to verify the token without communicating LTCA. In other words, an offline approach to issue pseudonymous credentials is that a PCA issues short-term certificates without contacting the appropriate LTCA. Regarding the fact that each LTCA issues a token for the vehicle to receive pseudonym certificates, there is no need for the PCA to ask LTCA about the permission, start-time and life-time. All the necessary information is encrypted, signed and stored in the token for the corresponding PCA. As a result, each PCA can decrypt and verify the token independently. The current solution is the offline approach.

Obtaining a limited set of pseudonyms requires two phases of communication. The first step is to communicate with an LTCA to receive a valid token, similar to the ticket in SAML[5] assertion. Figure 3.1 illustrates the communication between several vehicles and an LTCA to obtain a token in order to receive the pseudonym certificates.

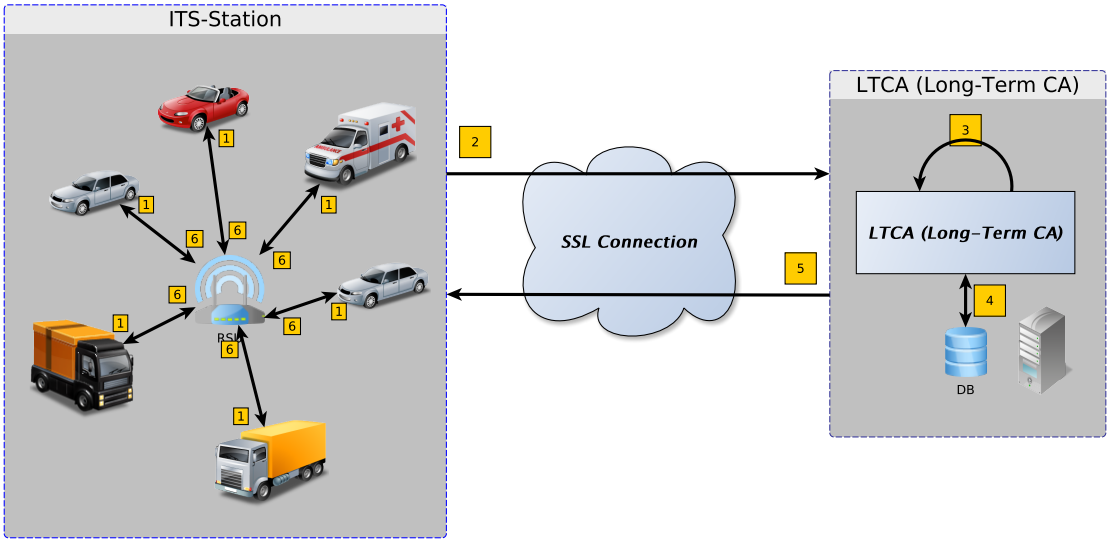

Figure 3.1: The Communication to Obtain Token

The second phase is to send the token to the corresponding PCA to obtain the pseudonyms. In this step, the vehicle first communicates with PCA while the PCA verifies the token without communicating with LTCA. The communication among ITS-Ss, a PCA and an LTCA in VPKI for acquiring new pseudonymous certificates is depicted in figure 3.2. As illustrated, there are 10 steps which have to be done in order to acquire the pseudonymous certificates.

To obtain short-term certificates, there is some information that PCA needs to know while this information has to be kept secret from the LTCA.[6] On the other hand, there is some other information that LTCA shall know while PCA is not supposed to know. The role of each authority has been divided based on the *"separation of role"* and *"separation of privilege"*[7] principles in order to achieve privacy. The reason to divide the protocol communication into two phases is mainly to protect user's privacy. In the first step, each vehicle is equipped with a token, which shows the authenticity and legitimacy of the

[5] Security Assertion Markup Language
[6] The reason is mostly to protect users' privacy.
[7] According to Bishop: *"The principle of separation of privilege states that a system should not grant permission based on a single condition"* [4].

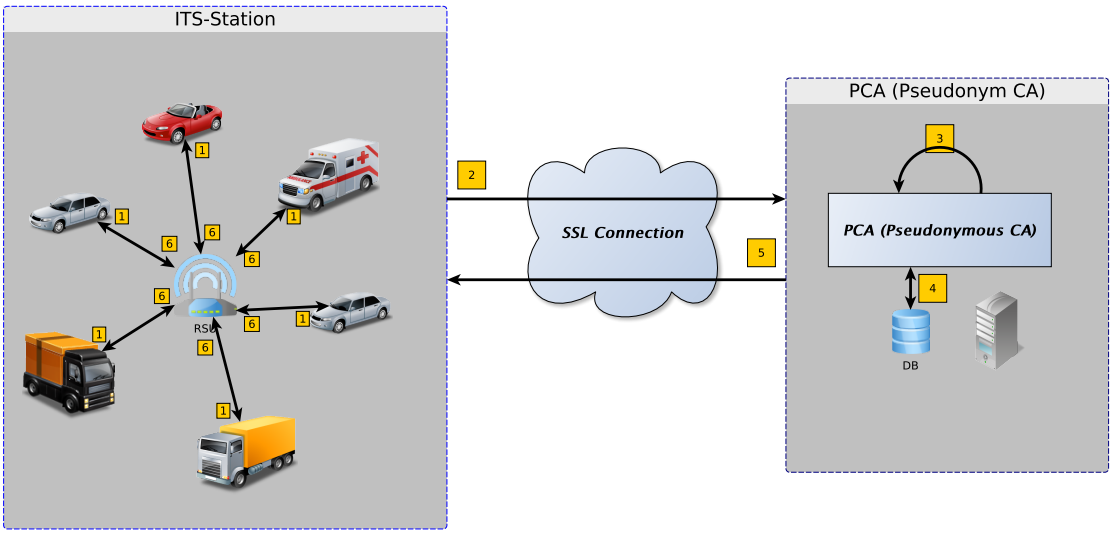

Figure 3.2: The Communication to Obtain Pseudonym Certificates

vehicle in the system. In the second phase, PCA can verify each vehicles authenticity and legitimacy using that token. As a result, no single authority can make a link between a long-term certificate and short-term certificates to identify the vehicle's real identity.

Figure 3.2 illustrates the communication between several ITS-Ss and a PCA. The first step (shown as #1 in the figure) is to establish a secure channel. There is an open-source toolkit called OpenSSL[8][22], used to establish an encrypted, integrity-protected and robust channel in order to communicate securely. Simultaneously, each vehicle generates a finite set of pseudonymous public keys and private keys, identified as K_V and k_V respectively. The symmetric keys for confidentiality and integrity are generated during the SSL tunnel establishment phase. It generates two secret keys: SK_{V-PCA}[9] and IK_{V-PCA}[10].

Before getting into the details of the communication protocol, there are some points that need to be emphasized here. First of all, each vehicle cannot use one of its pseudonymous certificates to establish a secure channel. Using the long-term certificate during the secure channel establishment infringes the vehicle's privacy since the PCA can figure out the real identity of the vehicle. Moreover, using the current valid pseudonym certificate will completely violate user's privacy since the PCA can make a link between different pseudonyms. Furthermore, the number of pseudonymous public keys and private keys depends on the capacity of communication, storage limitations on the vehicle side and other policies. It is obvious that the number of pseudonymous certificates does not really affect the correctness and accuracy of the protocol. To be more clear, the number of pseudonym certificates is encapsulated within the token and the LTCA determines the maximum number of them. No need to mention that the vehicle cannot cannot make a request more than the maximum number.

[8]http://www.openssl.org/
[9]The Session-Key between the vehicle and the PCA.
[10]The Integrity-Key between the vehicle and the PCA.

3.3.1 Token Format

Each token, issued by an LTCA, comprises the following fields:

{Token-Type, Token-Serial No., Token-Identifiable-Key, LTCA-Id, PCA-Id, Maximum Number of Pseudonym Certificates, Token Start-Time, Token Expiry-Time, Pseudonym Start-Time, Pseudonym Expiry-Time, Signature}

Token start-time and token expiry-time determine the life time validity of the token whereas pseudonym start-time and pseudonym expiry-time manifest the start-time and the expiry-time for the entire set of pseudonyms. PCA, in the second phase of pseudonym generation, divides the time slot to issue pseudonym certificates. In section 3.5.1, *Token-Identifiable-Key* which is used in the process of pseudonym resolution, is explained with great details. Section *Binding Token to the Pseudonym Certificate* in 3.5.1 describes in details.

To protect users' privacy, each token is not bound to a specific vehicle. In other words, a vehicle can share the token with another vehicle, not necessarily a legitimate one; as a result of this, the other vehicle can obtain pseudonym certificates and perform a sybil attack.[11] To mitigate such a threat, appropriate countermeasures should be applied. Section 3.5.1 provides additional details to prevent the threat. The assumption is that LTCA is a trusted authority and it never binds a faked long-term certificate in order to issue a token.

3.3.2 Pseudonym Certificate Format

Each pseudonym certificate, issued by a PCA, consists of the following fields:

{Serial No., Pseudonym Cert. Identifiable Key, Signer-ID, Valid-From, Valid-To, EC Public key, Signature}

Pseudonym Certificate Identifiable Key is used in performing pseudonym resolution. Section *Binding Token to the Pseudonym Certificate* 3.5.1 clarifies how it is involved in the process of resolving a pseudonym. EC public key determines the Elliptic Curve public key since Elliptic Curve Cryptography (ECC) is used in pseudonym certificates [3].

Having access to the pseudonyms, the real identity will not be disclosed while the authenticity and validity of the pseudonym can be completely verified. Moreover, in all the subsequent protocols, both time-stamp and nonce are used in order to mitigate Man-in-the-Middle and replay attacks. The purpose of using both is identical. However, the difference between using time-stamp and nonce is that using time-stamp, both parties have to be synchronized whereas using nonce, there is no need to keep both parties synchronized. Moreover, using nonce, the request and response are completely bound to each other while using time-stamp, the request and response are not completely bound to each other.[12] For the sake of completeness, nonce and time-stamp are both used in the forthcoming protocols.

[11] Sybil attack represents that an attacker masquerades another legitimate entity within the system, or masquerades multiple identities in the system. Further discussion is beyond the reference of this study. However, in the last chapter, conclusion and future direction 6.1, it will be discussed more.

[12] Using time-stamp, the sender and the receiver have to verify the time-stamp within a threshold which is not a completely trustworthy criterion. The sender and the receiver might not be synchronized; or, there might be some network delay. Further discussion is outside the extent of this report.

3.3.3 Pseudonym Certificate Life-Time Policy

To issue pseudonym certificates, there are two different policies to determine the start-time and life-time for each pseudonym. This can be performed on the LTCA side, which increases the workload of LTCA; or, it can be done on the PCA side. In the latter case, LTCA just determines the start-time and life-time for the entire set of pseudonyms while the PCA divides them for each pseudonym. As an example, suppose there are 4 pseudonyms as below:

$$\{K_V^1, K_V^2, K_V^3, K_V^4\}$$

Using the first alternative, LTCA sends the following information below to the PCA:

$$\{K_V^1, [t_0, t_0+\Delta]\}$$
$$\{K_V^2, [t_0+\Delta, t_0+2\Delta]\}$$
$$\{K_V^3, [t_0+2\Delta, t_0+3\Delta]\}$$
$$\{K_V^4, [t_0+3\Delta, t_0+4\Delta]\}$$

In the second alternative, LTCA sends $[t_0, t_0+N\Delta]\}$ to the PCA to specify the start-time and expiry-time for the entire pseudonyms and PCA divides the start-time and life-time for each pseudonym.[13] Having received the token from the vehicle, PCA divides the life-time for each pseudonym as follows:

$$\{K_V^1, [t_0, t_0+\Delta]\}$$
$$\{K_V^2, [t_0+\Delta, t_0+2\Delta]\}$$
$$\{K_V^3, [t_0+2\Delta, t_0+3\Delta]\}$$
$$\{K_V^4, [t_0+3\Delta, t_0+4\Delta]\}$$

Depends on the policy, either LTCA prepares the pieces of start-time and life-time for each pseudonym, or LTCA informs PCA the start-time and expiry-time for the entire request and PCA divides start-time and life-time for each pseudonym. In this project, the second policy is applied.[14]

The idea is that LTCA shall be informed about the period of validity for each pseudonym. In other words, LTCA does not know the pseudonyms for each real identity while it knows the validity period for the entire pseudonyms. It is similar to defense in depth where there exist two layers of defense. First, LTCA performs some validation and if it passes, the request goes to the PCA to be finalized.

3.3.4 Obtaining Token from LTCA

In the fist step, each vehicle sends the following message to LTCA to receive a valid token.

(msg #1) Veh –> LTCA:

$$SK_{V-LTCA}\{REQ\_TOKEN, \text{Long-Term Cert. Length, Long-Term Cert., LTCA's ID, PCA's ID, Nonce, TS, } \sigma_V^{LTC}(\text{ENTIRE MESSAGE}) \}$$

[13] N illustrates the number of pseudonyms for the request.

[14] In terms of implementation, the token format is simpler to specify only the start-time and the expiry-time for the entire pseudonym.

LTCA replies the vehicle by the following message: **(msg #2) LTCA –> Veh:**

$$SK_{V-LTCA}\{RES\_TOKEN, Token\ Size, Token, Max.\ No.\ of\ Pseudonyms, LTCA's\ ID, PCA's\ ID, Nonce{+}1, TS, Err.\ Code, EMD, \sigma_{LTCA}(ENTIRE\ MESSAGE)\}$$

3.3.5 Obtaining Pseudonym Certificate from PCA

The second phase is to send the token to the corresponding PCA.[15] The vehicle sends the following message to the PCA to obtain short-term certificates:

(msg #1) Veh –> PCA:

$$SK_{V-PCA}\{REQ\_PSEU\_CERT, Token\ Size, Token, LTCA's\ ID, PCA's\ ID, location, PseuCert.\ No., \{K_V^i\}, Nonce, TS\}$$

Upon reception of the message from the vehicle (identified as step 3 in the figure), PCA verifies the integrity, freshness and authenticity of the message, and stores the received pseudonymous credentials. In the case of any error or fault, it replies directly to the vehicle. Meanwhile, the PCA decrypts the token to verify its validity and authenticity.

PCA issues the pseudonym certificates for the vehicle and sends the final message to the vehicle:

(msg #2) PCA –> Veh:

$$SK_{V-PCA}\{RES\_PSEU\_CERT, LTCA's\ ID, PCA's\ ID, PseudCert.\ No., \{Pseu\_Cert\_SerialNo, PseuCert.\ Identifiable\ Key, PCA's\ ID, [t_i, t_i+\Delta], K_V^i, \sigma_{PCA}(ENTIRE\ PSEUDONYM)\}, Nonce{+}1, TS, Err.\ Code, EMD, \sigma_{PCA}(ENTIRE\ MESSAGE)\}$$

As the final step, the vehicle sends an acknowledgement to prove that pseudonyms are securely received and stored, or if an error happens:

(msg #3) Veh –> PCA:

$$SK_{V-PCA}\{ACK\_OK, Nonce{+}2, TS, Err.\ Code, EMD\}$$

To sum up, it is worth-mention to say that PCA can select some of the pseudonym public keys, and ask the vehicle to sign them to make sure that the vehicle has the corresponding private keys[16]. Authentication at the beginning suffices to authenticate the vehicle. However, PCA can perform this operation to reduce the probability of DDoS attack. LTCA and PCA remind defense in depth mechanism. Firstly, each vehicle is verified by an LTCA. In the second phase, PCA verifies its token to issue pseudonym certificates. The design here also reminds kerberos architecture while it differs in different aspects.

One of the most important issues is the linkability. A vehicle cannot request to obtain pseudonym certificates using the current valid pseudonym certificates since the PCA can make a link between different pseudonyms. That is why this protocol is divided into two phases. Using the token, PCA cannot make a link between different pseudonyms while it can verify and authenticate the vehicle.

[15] Usually, the corresponding PCA is normally the closest PCA to the vehicle. However, in this case, the corresponding PCA is the one that LTCA has encrypted the token for.

[16] This is considered an expensive operation since it adds two more steps to the protocol.

Above all, the mentioned protocol supports Perfect Forward Secrecy (PFS). Adding perfect forward secrecy is another important factor, considered during the design of the protocols in VPKI. Based on [28], perfect forward secrecy (PFS) means that the conversation between two entities has to be kept secure even if an eavesdropper records the entire encrypted conversation and he managed to break the long-term keys of both parties. The idea is to use some temporary keys as the one time session keys.

Furthermore, the current protocol supports escrow-foliage meaning that in the case that some totally benign and completely trustworthy organizations require both sides of communication to reveal their long-term private-keys, the prior encrypted communications would remain secret. To put it in another way, having access to prior knowledge of both parties long-term keys, decrypting the conversation is impossible by an eavesdropper [28].

An illustrative example of such situation is that suppose a vehicle sends his private key to PCA using a pre-established secure channel and PCA replies by a signed pseudonym certificate. This approach is problematic since an attacker may eavesdrop the channel and in the case of compromising the session key, the private key will be revealed. However, if the vehicle sends only the public key to the PCA and it replies with the signed pseudonym certificates, the only achievement of breaking the secure channel, by an intruder, is figuring out the certificate. Thus, prefect forward secrecy is considered in this attempt while it is not in the previous case.

As illustrated in the protocol details, based on the time-stamp or time of transaction, the real identities of the vehicles cannot be revealed. In other words, since there are many PCAs and less LTCAs within the domain of a secure VC system, the time-stamp is not a factor to help an attacker to figure out the real identity. If an attacker has access to the LTCA database, he cannot figure out the real identity using time-stamp and the *Token*. PCA may issue pseudonym certificates based on the time-stamps, valid in the tokens time-stamps validity period. Another issue is that pseudonym private keys have to be stored in the HSM so that a malware or a malicious entity cannot access the pseudonym private keys to sign a message. If pseudonym private keys are not kept secret, the non-repudiation requirement is not provided. Furthermore, the pseudonym certificates are stored on the OBU, On-Board Unit.

Another legal issue to consider is that how long the information regarding a vehicle shall be stored on PCA and LTCA databases. Further discussion on legal issues is beyond the scope of this research while it does not affect the correctness of the protocols. Everything depends on the policy of the secure VC system. Additionally, the other assumption is that PCAs and LTCAs are completely trusted companies or government companies. The idea to consider one extra trusted third party to store all the information of a vehicle is completely rejected. All the efforts has been done to separate the information into two places, based on the principle of *"separation of role"* and *"separation of privilege"*. Keeping all the information in one place is more probable to be compromised.

To sum up, it can be inferred that the protocol to obtain pseudonymous certificates is designed in a simple manner. The TLS protocol is used to perform mutual authentication and a secure channel is established between ITS-Ss, PCA and LTCA. Moreover, using symmetric and asymmetric cryptography, every message is integrity-protected using signed HMAC and an adversary cannot impersonate or forge messages. An attacker cannot do replay attack by sending older versions of messages. The secrecy of messages is protected using symmetric keys, created in the AKA phase of SSL. Even using faked

devices on the way cannot yield in Man-in-the-Middle attack since the certificates are being checked if they are signed by a correct RCA.[17] However, the only abuse which can be occurred is that an attacker can interfere the communication by jamming the traffic on the network. This pattern has been discussed in details in chapter 2, the adversary model, section 2.6.

3.4 How to Request the Latest Pseudonym CRL

When it comes to the revoked certificates in a secure VC system, there exist two kinds of CRLs: pseudonymous CRLs as well as long-term CRLs. The former includes the revoked pseudonymous certificates issued by a PCA and the latter is the revoked long-term certificates, issued by an LTCA. There is no reason for a vehicle to deal with long-term certificates.[18] As a result, the main focus of this protocol is on pseudonym CRL.

There are some reasons to revoke pseudonymous certificates such as: vehicle robbery, vehicle crash in accidents, vehicle misbehaviour and breaking the law, vehicle breakdown, vehicle insurance breaking law, and at the end of vehicles' life cycle (junked vehicles)[19].

To prevent faulty nodes from damaging the secure VC system, the certificates of faulty nodes should be revoked by a legitimate trusted authority. The decision to revoke a certificate is made by the CA due to the administrative or technical reasons [1]. In this case, PCA is the responsible authority to revoke pseudonym certificates. The most common and basic mechanism to propagate revoked certificates among the nodes within a VC system is using Certificate Revocation Lists (CRLs) [1]. The main reason to quickly disseminate the lists of revoked certificates is to keep the system operating always in a secure state since certificate revocation is deemed necessary to evict illegitimate and faulty nodes from the system. It also excludes compromised nodes, not to allow attackers to violate the system security policy [6].

When it comes to the CRLs, it is well perceived that this is a broad area. There are many details in terms of disseminating certificate revocation lists in a VC system. They are not expected to be deeply discussed here since they are outside the sphere of reference. A detailed discussion of CRLs are addressed in [6].[19] One of the concepts in the area of certificate revocation list distribution in a VC system is *"revoked foreigner certificates"*. It is a special kind of certificate, issued to the foreigner visitors, to travel in a region outside of their realm. This is also outside the scope of this research.

As suggested in [6], each CRL shall be disseminated once a month, for example.[20] There are some approaches to diminish the large size of CRLs, such as using Δ-CRLs.

[17] MitM is possible if a vehicle is compromised and faked CA's certificates are stored on the vehicle.

[18] All the transactions by the vehicles is done using pseudonym certificates. Long-term CRL is used by authorities such as PCAs and LTCAs. This is not a matter of focus in this protocol.

[19] [6] provides a detailed description on *"Regional CRLs"*, *"CA Collaboration"*, *"Multi-RSU CRL Distribution"*, *"Randomized low-Rate Broadcast Distribution"*, *"Segmented, Erasure-Coded Protected, Secure CRLs"*, *"Minimal RSU-CA and No RSU-RSU Interactions"*, *"Randomized, Low-Rate Broadcast Distribution"*, *"CRL Construction"* and *"Foreign CRL"*.

[20] As exemplified in [19], some privileged entities can be used in order to disseminate some emergency messages. For example, emergency vehicles, police and buses are the trusted entities within the domain of a VC system to broadcast the official messages. They can be used in order to inform other vehicles about a recently published CRL.

The idea is to append recently revoked certificates to the latest CRL to inform nodes. To put it in another way, Δ-CRL is considered a faster approach to distribute revoked certificates only if the full CRL was received beforehand [6]. Using Δ-CRLs in an *"error-prone, highly volatile, often disconnected VC environment"* is not a desirable and appropriate solution [6]. There are some approaches other than certificate revocation lists. One approach is to send a number which is interpreted as a threshold number and all the other certificates with identities below that number are invalid. The dominant approach here is to use CRLs. In chapter 5, the performance of such approaches are discussed.

Two kinds of certificates exist in the domain of a secure VC system: short-term certificates as well as long-term certificates. Pseudonymous certificates shall be revoked by a PCA while long-term certificates shall be revoked by an LTCA. As a result, each vehicle is supposed to ask an appropriate PCA to receive the latest CRL of pseudonymous certificates; whereas, it is expected to ask an LTCA to retrieve the CRL of long-term certificates. The idea of retrieving all the CRLs from only one authority (PCA, or LTCA, or another trusted authority) is highly rejected since it has many disadvantages while it has less benefits. The standard emphasizes on separating short-term revocation lists and long-term revocation lists [20]. Integrating both revoked pseudonym certificates as well as revoked long-term certificates into one list to be disseminated has the following shortcoming:

- *High workload on either PCA or LTCA*
- *Single point of failure*
- *Single point of contact*
- *Difficulties in terms of management*

Integrating these two CRLs is not efficient since pseudonym CRL is useless for every authority and long-term CRL is pointless for the vehicles. It is important to emphasize that since PCA issues the pseudonym CRLs, it has them while it does not have all the pseudonym CLRs issued by all the PCAs. Pseudonym CRLs are used in the process of obtaining pseudonym CRL since each vehicle has to sign the request using its current valid pseudonym certificate. Since an LTCA is not responsible to provide services for pseudonym certificates, sending queries to the LTCA to achieve revoked pseudonym certificates increases the workload on LTCA while it affects the efficiency as well. Moreover, the goal was to separate the role of the PCA and the LTCA, not to allow one single node to access the whole information. Collecting both long-term and short-term revoked certificates in one entity might violate the vehicles' privacy. Another argument is that each authority would be considered as a single point of contact or failure, in the case of combining two CRLs. As a result, if one authority is not reachable, the CRLs cannot be fetched at all and it is problematic. Additionally, it makes it complicated since there is a need for extra tasks to combine these two lists. Thus, according to the standard, the lists are disseminated separately.

For the sake of completeness, it is also worth-mentioning that to revoke the RCA certificate, measures have to be applied in order to keep the system secure. Usually RCA certificates are valid for a longer period of time while other certificates are not valid as long as RCAs' certificates life time. There must be some measures in order to replace and update the RCAs' certificates. In some circumstances, an RCA's, PCA's, LTCA's and PRA's certificate might be compromised and subsequently, it can threaten the whole

system. In the case that an RCA's certificate is not trust-worthy anymore, an administrator can revoke the certificate using its identity. As stated in IEEE 1609.2 v2,

> "the revocation of RCA certificates has to be done in extreme cases like a successful attack or some other catastrophic situations. So when the RCA certificate is compromised or is not trust-worthy anymore (for some reason), can be manually revoked by the administration and afterwards, the cert-ID of the revoked certificate is attached to the CRL and finally signed by the RCA. With incremental serial number, the receivers meaning PCA, LTCA and ITS-Ss can figure out if the currently stored RCA certificate is the latest version or it has to be exchanged with other and a new version of RCA certificate [19]."

Further discussion is irrelevant and out of the scope of this research project.

One of the most controversial issues here is that what happens if a long-term certificate is revoked and how the corresponding pseudonym certificates are revoked. There must be some mechanisms such as reverse pseudonym resolution. Having revoked a long-term certificate, the corresponding short-term certificates have to be revoked without disclosing the real identity of the pseudonyms to the PCA. Further discussion is irrelevant since they are beyond the reference of this study. It will be discussed more in the future works in section 6.1.

In other words, disseminating the CRLs is dedicated not only to the revoked long-term certificates, but also to the revoked pseudonymous certificates. As mentioned in [6], the only difference between these two kind of certificates is the size of CRL. If a vehicle, equipped with multiple pseudonymous certificates, which has to be evicted, all of its unexpired pseudonymous certificates has to be revoked and added into the CRL [6]. Interestingly, IEEE 1609.2 proposes using a large pool of pseudonyms, shared among vehicles. Each vehicle, in case of need, picks a small subset of the pseudonyms credentials and uses them accordingly [6].

The idea is to keep the CRL size as low as possible so that the list of revoked certificates can be distributed among the entities within the system in a reasonable time; it might take in the range of minutes, as stated in [6]. An important factor here, which needs to be considered as well, is foreign vehicles. As expressed in [6], foreigner visitors' certificates are validated and subsequently, new certificates named short-term Foreigner Certificates (FCs) are issued for the visitors. It goes without saying that in the case of being revoked, they will be included in the successive CRL to inform all the entities within the region [6].

The details of detecting nodes misbehavior or compromised nodes are outside the confines of the thesis project. The assumption is that they are being detected in a way. The discussion here is how vehicles shall commence the protocol to retrieve the latest and up-to-date CRLs from the corresponding CAs. There are some approaches to inform entities about the recent version of CRLs. The most common and basic approach is that each vehicle regularly requests for CRLs and receives the latest CRL [6]; furthermore, CAs might broadcast some information about releasing the latest versions of the CRLs [6].

In the next section, pseudonym CRL format is discussed.

3.4.1 Pseudonym CRL Format

Each pseudonym CRL, issued by a PCA, contains the items below:

{Pseu-CRL Serial No., CRL Version, PCA-Id, Number of Revoked Pseu-Cert., Revoked Pseu-Cert. Serial No., Time-Stamp, Signature}

Revoked Pseudonym Certificate Number identifies the number of revoked pseudonym certificates in this version of CRL. Revoked Pseudonym Certificate Serial Number shows the serial number for the revoked pseudonym certificates. No need to mention that pseudonym CRL includes the non-expired revoked pseudonym certificates, issued by a specific PCA-Id. The expired pseudonym certificates are not shown in the CRL.

3.4.2 Short-Term CRL Protocol

Figure 3.3 demonstrates the communication between several ITS-Ss and a PCA. Each step is numbered to identify the communication phases while messages are going back and forth to retrieve the CRLs for pseudonym certificates.

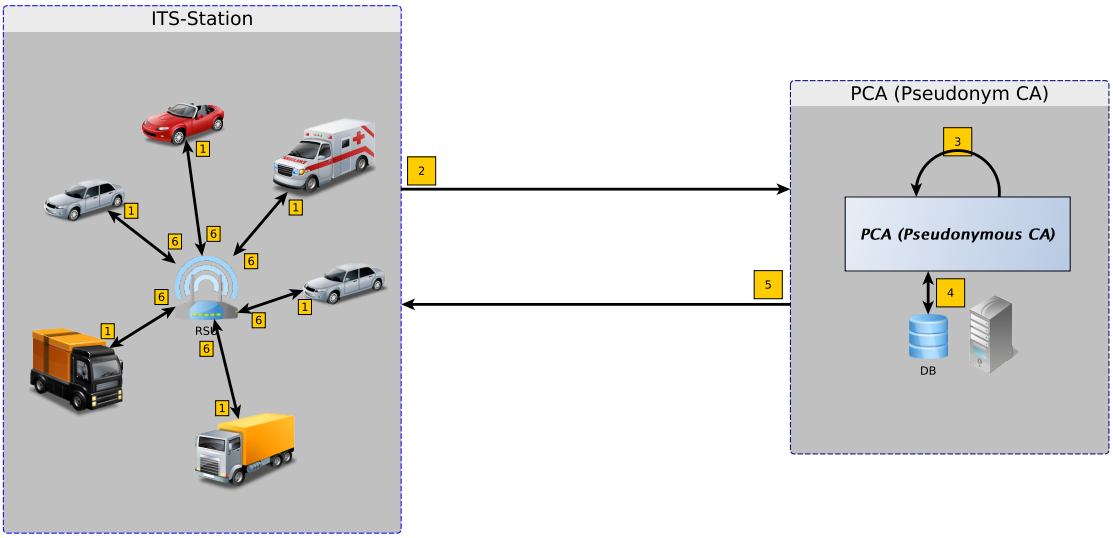

Figure 3.3: The Communication to Obtain Pseudonym CRL

To obtain the latest version of pseudonym CRL, there is no need to establish a secure SSL channel. CRLs are considered the publicly available information and each vehicle is legitimized to obtain them. Receiving the pseudonym CRL from PCA requires the vehicles to be a legitimate ones within the system. This is achieved by signing the requests using the current valid pseudonym certificate. Moreover, pseudonym CRLs are signed by PCA and they can be verified by the recipient. Thus, to obtain the pseudonym CRL, a new request is encapsulated to be sent towards the PCA. This step is identified as step 2 in figure 3.3.

To acquire the CRLs, a request has to be sent towards a PCA, as below:

(msg #1) Veh –> PCA:

{REQ_PSEU_CRL, Current CRL Version, PCA's ID, Region-ID, PseuCert. Length, $Cert_{PCA}(K_V^i)$, Nonce, TS, $\sigma_V^{k_v^i}$(ENTIRE MESSAGE)}

Upon receipt the message, PCA verifies the current pseudonym certificate validity as well as verifying the signature on the entire message. Moreover, it checks the pseudonym

CRL to see if this pseudonym is not revoked. Thus, the authenticity of the sender and the correctness of the request is verified.

In the next step, PCA retrieves the latest CRL for pseudonym certificates from its local database and encapsulates them. Not to mention that the CRLs are inherently signed by PCA and integrity is innately provided. Message 2 is the response from the PCA to the vehicle, illustrated in details below:

(msg #2) PCA –> Veh:

{RES_PSEU_CRL, PCA's ID, CRL Size, Pseudonym CRL[21]*, Nonce+1, TS, Err. Code, EMD,* σ_{PCA}*(ENTIRE MESSAGE)}*

In the last phase, the vehicle retrieves the latest pseudonymous CRL. The error code and error message description are also checked, in the case of happening. It is possible that the messages are not being sent in one packet and as a result, the CRLs are separated into several packets with different sequence numbers. This does not affect the correctness of the protocol. There is a chance also to send them by adding redundancy, using erasure codes, such as using Robin's algorithm in a secure message transmission [29], to improve the efficiency.

An important fact which needs to be considered is the CRL size, which plays a critical role, in terms of time, when it comes to the performance. If the secure VC system works in a normal state, the number of revoked pseudonyms are much bigger than the number of revoked long-term certificates. This is totally outside the confines of this thesis project; however, when it comes to the evaluation and optimization, it has to be considered.

3.5 Pseudonym Resolution

The idea of pseudonym resolution is to identify the real identity of a pseudonym. As illustrated in figure 3.4, police, PRA[22], PCA and LTCA are all involved to perform pseudonym resolution. The important fact is to determine the real identity of a pseudonym. What is not a matter of focus here is that how and in which circumstances the pseudonym needs to be resolved by an authority.

This protocol is designed to identify the real identity of a pseudonym certificate, issued by a PCA. An authority such as police is able to query the PRA to fetch and retrieve the real identity for a specific pseudonym. To do so, PRA first communicates with PCA to retrieve the token-Id and subsequently, it queries LTCA to identify the real identity of that token. It is assumed that a secure channel has been already established between PRA-PCA and PRA-LTCA.

As pointed out in token format in section 3.3.1 and pseudonym certificate format in section 3.3.2, there are two field identified as: Token-Identifiable-Key and PseuCertIdentifiableKey. Before getting into the details of the protocol, it is worth to explain the details of them in the token and pseudonym certificate formats.

[21] Based on the format discussed in section 3.4.1

[22] Pseudonym Resolution Authority: an authority, certified by RCA, identified as: $Cert_{RCA}(PRA)$.

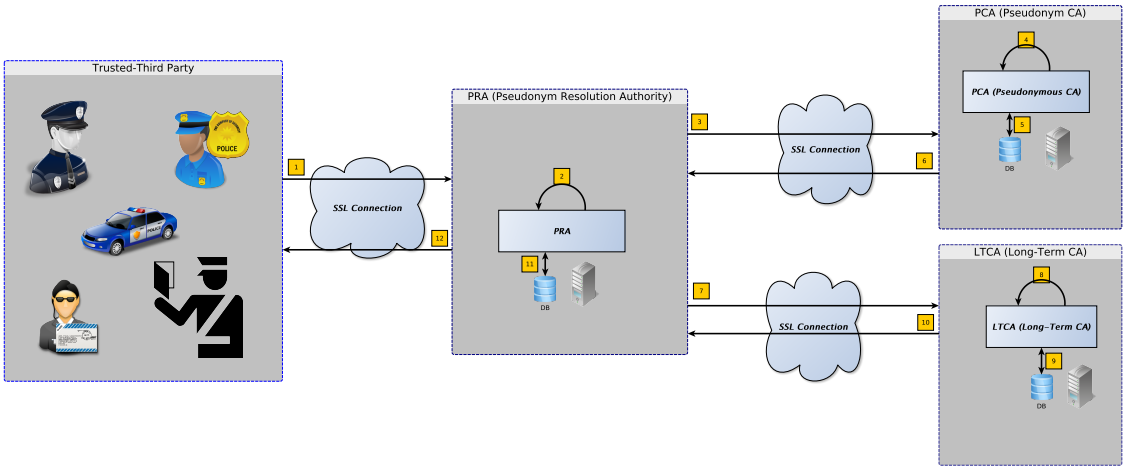

Figure 3.4: The Communication to Perform Pseudonym Resolution

3.5.1 Binding Token to the Pseudonym Certificate

One of the most challenging issues when it comes to resolving a pseudonym is that how the PRA can verify the integrity of a token, assigned to a vehicle. In other words, how PRA can recognize if PCA has resolved the pseudonym certificate with a faked, or a real token. The idea is to keep user's private information secure while it has to be verifiable. Thereby, when it turns out to issue a token, each token contains a hash digest of {vehicle's long-term certificate serial number, time-stamp and a nonce}.[23] Time-stamp and nonce play the role of salt to make it difficult from recovering the serial number. Thus, TokenIdentifiableKey is calculated as below:

$$TokenIdentifiableKey = hash(Vehicle\ Long\text{-}Term\ Certificate\ Serial\ No.\ ||\ Time\text{-}Stamp\ ||\ Nonce)$$

Taking into consideration, each pseudonym certificate, issued by a PCA, consists of the hash digest of {TokenIdentifiableKey, Pseudo-Public Key, time-stamp, nonce}, as follows:

$$PseuCertIdentifiableKey = hash(Token\text{-}Identifiable\text{-}Key\ ||\ Pseudo\text{-}Public\ Key\ ||\ Time\text{-}Stamp\ ||\ Nonce)$$

Having stored PseuCertIdentifiableKey inside each pseudonym certificate, the PCA cannot resolve a pseudonym certificate with a false token-Id since it will be recognized by LTCA in the next step. However, in the case of resolving a wrong long-term certificate by LTCA, it will be recognized by police authority. Due to the assumption, all the issued tokens for a vehicle are stored on the vehicle's HSM. If the police resolves a pseudonym certificate after an accident, LTCA cannot resolve a faked long-term certificate since the corresponding token should have been stored on the vehicle's HSM. Based on the assumptions, tokens are stored in the HSM and cannot leave the storage. As a result, the vehicle cannot delete some of the tokens in the case of accident or deny having a token later on. All in all, PCA and LTCA are not able to resolve a false token and a faked long-term certificate for a pseudonym certificate.

[23] SHA256 is used to calculate the hash digest.

To sum up, PRA queries PCA as well as LTCA to figure out the real identity of a pseudonym in order to perform pseudonym resolution. As stated, PCA cannot reply with a faked response since it will be recognized by LTCA in a later phase. Moreover, LTCA is not able to generate a bogus answer and subsequently, violate the system security policy. The assumption is that LTCA is a trusted entity and it does not bind a faked certificate at the beginning phase to issue a token.

In the next section, the details of the protocol to accomplish pseudonym resolution is explained.

3.5.2 Pseudonym Certificate Resolution Protocol

The first step to perform pseudonym resolution is sending a request from an authority, like police, to the corresponding PRA. No need to mention that it is assumed that secure channels are established between Police-PRA, PRA-PCA as well as PRA-LTCA. The message consists of the pseudonym certificate, which has to be resolved, the identification of police and the PRA, nonce, time-stamp as well as a signature using the police's long-term private key. The following message is sent to PRA:

(msg 1) Police-Authority –> PRA:

$$SK_{Police-PRA}\{REQ\_PSEU\_RESOLUTION,\ PseuCert.\ Length,\ Cert_{PCA}(K_v^i),\ Police's\ ID,\ PRA's\ ID,\ Nonce,\ TS,\ \sigma_{Police}(ENTIRE\ MESSAGE)\}$$

Having received the pseudonym resolution request, PRA queries PCA to retrieve the corresponding Token for that pseudonym. The pseudonym certificate issuer's identification determines the corresponding PCA. As a result, the following message is sent to PCA:

(msg 2) PRA –> PCA:

$$SK_{PRA-PCA}\{REQ\_TOKEN\_ID,\ PseuCert.\ Length,\ Cert_{PCA}(K_v^i),\ PRA's\ ID,\ PCA's\ ID,\ Nonce,\ TS,\ \sigma_{PRA}(ENTIRE\ MESSAGE)\}$$

PCA shall resolve the pseudonym certificate by sending the corresponding token for the pseudonym. Since the pseudonym certificate is bound to the token, as explained in 3.5.1, PCA cannot resolve a false, or faked token. In the case of resolving incorrect, LTCA will recognize and report it. As a result, PCA verifies the authenticity of the PRA, resolves the corresponding token and responds as below:

(msg 3) PCA –> PRA:

$$SK_{PRA-PCA}\{RES\_TOKEN\_ID,\ PseuCert.\ SerialNo,\ Token\ SerialNo,\ Token\ Identifiable\ Key,\ PCA's\ ID,\ PRA's\ ID,\ LTCA's\ ID,\ Nonce{+}1,\ TS,\ Err.\ Code,\ EMD,\ \sigma_{PCA}(ENTIRE\ MESSAGE)\}$$

In the third step, PRA sends a request, including the resolved token, to LTCA to retrieve the corresponding vehicle's long-term certificate. Thus, it encapsulates the following packet and sends it to the LTCA:

(msg 4) PRA –> LTCA:

$$SK_{PRA-PCA}\{REQ\_REAL\_IDENTITY, Token\ SerialNo, Token\ Identifiable\ Key, PRA's\ ID, LTCA's\ ID, Nonce, TS, \sigma_{PRA}(ENTIRE\ MESSAGE)\}$$

LTCA shall first verify the authenticity of the PRA. Afterwards, LTCA finds the corresponding long-term certificate in its local database and calculates TokenIdentifiableKey[24]. If the TokenIdentifiableKey matches to the one, which has been stored in LTCA's database, it shows that PCA has resolved the pseudonym correctly. Thereby, it responds the PRA as below:

(msg 5) LTCA –> PRA:

$$SK_{PRA-LTCA}\{RES\_REAL\_IDENTITY, Token\ SerialNo, Veh.\ Long\text{-}Term\ Cert.\ SerialNo, VLTC\ Size, Veh.\ LTC, LTCA's\ ID, PRA's\ ID, nonce{+}1, TS, Err.\ Code, EMD, \sigma_{PCA}(ENTIRE\ MESSAGE)\}$$

Finally, PRA retrieves the resolved long-term certificate for the pseudonym. It stored the pseudonym certificate, the corresponding token and the long-term certificate in the database for the upcoming requests. It then encapsulates the message below and sends the resolved long-term certificate to the police authority. The message contains: the pseudonym certificate, the identification for vehicle long-term certificate, the LTCA's Id, the PCA's Id, the PRA's Id, incremented nonce by one, as well as the time-stamp. As before, the error code and the error message description are also sent to the police, in the case of happening.

(msg 6) PRA –> Police-Authority:

$$SK_{Police-PRA}\{RES\_PSEU\_RESOLUTION, PseuCert\ Length, Cert_{PCA}(K_v^i), Veh.\ Long\text{-}Term\ Cert.\ SerialNo, VLTC\ Size, Veh.\ Long\text{-}Term\ Cert., PRA's\ ID, LTCA's\ ID, PCA's\ ID, Police's\ ID, Nonce{+}1, TS, Err.\ Code, EMD, \sigma_{PRA}(ENTIRE\ MESSAGE)\}$$

At the end, police retrieves the long-term certificate for the resolved pseudonym. To complete the transaction, it responds an acknowledgement to PRA to demonstrate that the resolved long-term certificate is successfully received and stored. The message below is sent to the PRA as the final acknowledgement:

(msg #7) Police-Authority –> PRA:

$$SK_{Police-PRA}\{ACK\_OK, Police's\ ID, PRA's\ ID, Nonce{+}2, TS, Err.\ Code, EMD, \sigma_{PRA}(ENTIRE\ MESSAGE)\}$$

To sum up, it is inferred that resolving a pseudonym never infringes the vehicles' privacy. First of all, by resolving a pseudonym, none of the PCA, LTCA, PRA, and the police can make a link among different pseudonyms. In other words, resolving a pseudonym certificate reveals the corresponding long-term certificate for that pseudonym and no single entity can determine all the related pseudonyms for a vehicle. Furthermore, performing pseudonym resolution shall be done in some specific circumstances which have to be determined by the law. Further discussion is out of the scope of this research.

Although PCA cannot resolve a faked or incorrect token for a pseudonym certificate, LTCA is able to resolve a faked long-term certificate. PRA is not able to figure out if LTCA resolves the token incorrectly. Based on the assumptions, tokens are stored on the HSM. However, a mechanism shall be applied to prevent LTCA from deviating the system policy. In the future works, it will be discussed more.

[24] 3.5.1 explains binding token to the pseudonym certificate in details.

Chapter 4

Design and Implementation

4.1 Introduction

Having designed the protocols and the infrastructure of a VPKI, they have to be implemented and deployed in order to be utilized and evaluated. From security viewpoint, implementation is considered vital since a simple vulnerability can violate the system security. In other words, an attacker could leverage a bug in implementation to hack into the system, or compromise a vulnerability. To be more precise, software defects consist of implementation *bugs*[1] as well as design *flaws*[2] [26]. Roughly speaking, half of the software defects are design-level flaws while the rests are implementation-level bugs [26]. Taking into consideration, implementation of such a critical infrastructure requires a lot of efforts from security perspective.

In this chapter, a brief discussion on the advantages of open-source software for this project is examined. Later on, the criteria to select the most suitable open-source implementations of PKIs are mentioned. In the next section, the selected libraries and the source-code implementations to be used in VPKI are stated. To perform the experiment, there is a need to set up a lab; the experimentation setup will be illustrated to show how the evaluation will be done. The achieved results also pinpointed afterwards in order to show the result of protocol implementations in the VPKI. The results will be useful to perform optimization and subsequently, enhance the performance.

4.2 Open-Source Software Advantages

To deploy and implement a VPKI within the domain of a secure vehicular communication system, free and open-source software (FOSS) is selected in order to implement and deploy the VPKI. FOSS holds many advantages regarding the current situation, the practical problem as well as the limitations of this project. Some of the significant reasons behind choosing FOSS are as follows:

[1] An implementation-lever software problem such as buffer overflows [26].
[2] A design-level software problem such as Microsoft Bob [26].

Security

Security of open-source software and proprietary software have been always a challenging issue within the worldwide software communities. FOSS always claims that the availability of the source-code to the public makes it more secure than the proprietary one. Thereby, everyone is able to scrutinize the source-code, test the software and as a result, find the existing bugs or flaws. On the other hand, proprietary software can be analyzed, scrutinized and tested by a limited number of experts within the organization. Based on many-eyes principle[3], the bugs and flaws are more likely to be discovered and fixed in open-source software. The reason is that the source-code is publicly available, and more experts are able to figure out bugs or flaws. Moreover, since they are free of cost, everyone can freely use and customize them.

Cost

No need to mention that open-source software and commercial or proprietary software are not comparable in terms of cost. This is also considered as an important factor when it comes to implement and customize the software for the purpose of VPKI.

Auditability

The fact is that everyone is not able to audit the source code of a closed-source and proprietary software[4] while a limited number of experts within the organization have access to do so. When it comes to FOSS, auditability is an outstanding feature since the source-code is publicly available.

Customizability

This feature is also a noticeable criterion since the aim is at customizing the currently available source-code of the open-source standard PKI in order to deploy a VPKI. In other words, the idea is to customize the implemented functions of the open-source PKI, modify them for the purpose of a vehicular communication system, and implement more functionalities regarding necessary security features of the VPKI.

Flexibility

Due to the fact of integrating the VPKI into the rest of the secure VC system, it is more flexible not to rely on a specific tool, software or hardware. In other words, since the source code is available, it can be executed and run on any system with no limitations in terms of support. The flexibility feature also makes it easier to modify, upgrade and customize the source-code at a later point.

[3]Many-eyes principle states that since the source-code of the software is publicly available to everyone, the probability to figure out the bugs in implementation or the flaws in the design is much higher. It goes without saying that not everyone really scrutinizes the open-source software; however, the rule of thumb says that experts in software communities scrutinize the code, analyze it and test it under different conditions. They certainly never provide a full security assurance while they do their best to fix the vulnerabilities.

[4]Non-disclosure agreement (NDA) is a reasonable and legal approach to sign a contract in order to audit the source code. This is not always an easy, straightforward and available solution for everyone, though; it has its own limitations.

4.3 The Open-Source PKI, A Comparison

To implement a PKI for a secure VC system, different open-source implementations of the standard PKI were analyzed and evaluated. The main criteria to select an implementation are:

- *The similarities of the VPKI features and requirements to the existing implementations*
- *Implementation language, compatibility with the rest of the project*
- *Completeness of documentation*
- *General support*
- *Support crypto-libs, SSL, LDAP and other services*
- *Easy to use*

The most significant criterion is the similarities of the required features within the domain of VPKI to the current implementations of the standard PKI. Based on the necessary functional and security requirements, which have been discussed in details in chapter 2, the implementation of each open-source PKI is analyzed to consider which one is mostly closed to the VPKI requirements. The implementation language is also another important challenge for the sake of compatibility to the rest of the project. Implementation language in C/C++ has an advantage since the rest of the project is in C/C++. Furthermore, the completeness of the documentation is considered important since it shows how much the development team have noticed documentation; moreover, it could be helpful for customizing and utilizing the source-code. Besides, general support illustrates if the project is still going on, or it has been already stopped. Supporting different cryptographic libraries, and being easy to use are other criteria in order to select an implementation for the purpose of VPKI.

A comprehensive and thorough analysis on different implementations shows the most suitable open-source implementation for VPKI. There exist many different implementations of the standard PKI in different languages. In the following, each of them is briefly discussed.

OpenCA[5]

OpenCA is one of the implementations of public key infrastructure to provide credential management for the standard PKI. It has not been finalized yet while the efforts to improve it, is still going on. It provides a comprehensive libraries while it uses the libraries of OpenSSL[6] [22], openpki and OpenLDAP[7] [30] to achieve more security functionalities. Furthermore, it is released with an Apache-style license [31]. The OpenCA guide is available in [23]. The details of OpenCA project are highlighted in [32]. OpenCA is written in C, and the documentation is also quite good. All in all, OpenCA is used as the skeleton of VPKI.

Mozilla Open Source PKI Projects[8]

As illustrated in [24], Mozilla is equipped with comprehensive libraries, which provides the functionalities of the standard PKI. It is equipped with two libraries called, Network

[5] http://www.openca.org/
[6] http://www.openssl.org/
[7] http://www.openldap.org/
[8] http://www.mozilla.org/projects/security/pki/

Security Services (NSS) and the Personal Security Manager (PSM). The former is a security module to provide libraries to support certificate and key management, SSL, S/MIME and cryptographic token support [24]. The latter is a set of libraries to support SSL, PKCS, S/MIME, X.509 v.3 certificates and many other features [24]. The libraries are written in C and it has complete and comprehensive documentations as well.

EJBCA[9]

As stated in [33], EJBCA is an open-source implementation of standard PKI, written in Java/J2EE. It plays the role of a CA and its main focus is on a scalable, flexible and platform independent CA/RA [33]. The documentation of the implemented functionalities are also quite good. Moreover, it consists of a SignServer LiveCD[10], considered as the two open-source projects called EJBCA as well as SignServer [34].

pyCA[11]

pyCA is another implementation of PKI, using python language, and it provides a WWW interface to the CA. It also provides several libraries for security while it uses OpenSSL as the mechanism for establishing a secure tunnel [35]. It has neither good documentation nor good support. Furthermore, the python language is also a constraint for this project.

PERMIS[12]

Permis provides necessary functionalities for X.509 Attribute Certificates, SAML attributes, and other necessary features for privilege management and authorization policies [36]. In other words, PrivilEge and Role Management Infrastructure Standards (PERMIS),

> "provides a cryptographically secure privilege management infrastructure (PMI) using public key encryption technologies and X.509 Attribute certificates to maintain users' attributes [37]."

Other Open-Source Implementations

There are some other open-source implementations of PKI like: Public-Key Infrastructure (X.509) (pkix)[13] [38], Oscar[14] [39], Jonah, MISPC and Open Source PKI (OSPKI)[15]. For the time being, no implementation or source-code of them is available and it seems that such projects are obsolete. They are not supported anymore and subsequently, these alternatives are rejected.

Having discussed in details about the different implementations, OpenCA is chosen as the base library and implementation. In the following, the selected libraries, used to implement the vpki, are discussed.

[9] http://ejbca.sourceforge.net/userguide.html
[10] http://wiki.ejbca.org/livecd
[11] http://www.pyca.de/
[12] http://sec.cs.kent.ac.uk/permis/
[13] http://datatracker.ietf.org/wg/pkix/charter/
[14] http://srg.cs.uiuc.edu/Security/nephilim/kerberos.html
[15] http://ospkibook.sourceforge.net/

4.4 Selected Libraries

Designing and implementing a vehicular public key infrastructure consists of using many libraries in terms of cryptographic libraries, communication, storing information, as well as serialization and deserialization. The idea of this project was to use the open source implementation of PKI, called OpenCA, as the main infrastructure. OpenCA is installed on all the servers, as shown in figure 4.1. To perform the required functionalities of the system, several libraries are used in order to accomplish functional and security requirements. In the following, each of them are described.

When it comes to cryptographic functionalities, the most well-known library, OpenSSL, is used. OpenSSL [22] is an open-source implementation of cryptography libraries, including SSL/TLS as well as well-known cryptographic algorithms. In this project, all the cryptographic features, including signing and verifying signature for Elliptic Curve algorithm (ECDSA), X.509 certificate verification, digest algorithm (SHA-256) as well as signing and verifying using private key and public key are used from OpenSSL library. [16]

One of the most important part of the project is to connect all the servers to communicate with each other. Each authority is installed on one server and the clients, either the vehicles or the police authority, make a request to the servers. The communication protocol is equipped with boost serialization library, in order not to be dependent on a specific technology. In other words, boost serialization library is used for serializing and deserializing data structure to a specific format[17]. Accordingly, the application is totally independent of the communication protocol. There are many technologies to establish the connection such as *SOAP*, *Restful*, *cgi*, *RPC* and *socket*. Based on the limitations of this project[18], RPC was selected as the communication technology and *xmlrpc* is selected as the library to connect vehicles to the servers. As a result, libxmlrpc [40] is used. Moreover, OpenCA uses CGI, as a technique to respond the client's query. This project is also equipped with CGI scripts to be executed by Apache web-service.

Owing to the fact that OpenCA uses MySQL to store information in database, VPKI is provided with MySQL to store information in the database. Since OpenCA is installed on each server, one table is created for each authority to store and retrieve information.[19]

4.5 Experimentation Setup

To accomplish the experiments, PCA, LTCA as well as PRA were installed on three different servers. Figure 4.1 illustrates how to perform the experiments. As shown in the figure, one server is dedicated to each authority while the clients send their requests to the appropriate servers. To obtain a token, each client sends its request to LTCA whereas obtaining pseudonym certificates requires the clients to send their requests to PCA. To perform a pseudonym resolution, the client transmits its request to the PRA. As displayed in the figure, all the servers are located in the same domain. *OpenCA*, *OpenSSL*, *Apache*, *Mysql* as well as the rest of the prerequisite packages were installed on all the servers.

[16] The add-on libraries from openssl to the project are *libssl* and *libcrypto*.
[17] libboost_serialization library is added to the project.
[18] Implementation language as well as integrating the vpki into OpenCA are the main two constraints.
[19] *libmysqlclient* library is used on each server.

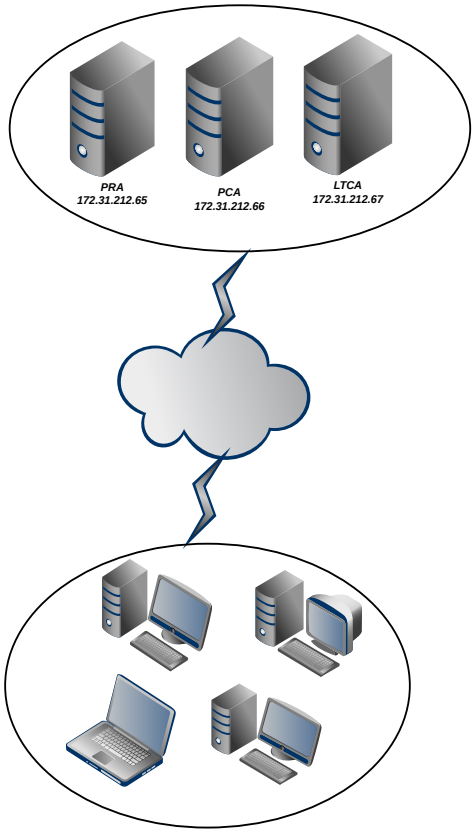

Figure 4.1: Network Experimentation Setup

4.6 Design and Implementation of VPKI

To implement the protocols, discussed in chapter 3, a library, called vpkilib, is written. This library contains 5 different modules, namely:

- *pseudonym_token*
- *pseudonym_certificate*
- *pseudonym_crl*
- *pseudonym_resolution*
- *dataBaseManagement*

pseudonym_token module is dealing with issuing token, used by LTCA and the vehicles to obtain a token. *pseudonym_certificate* and *pseudonym_crl* are used by PCA as well as the vehicles to obtain pseudonym certificates and pseudonym CRLs. *pseudonym_resolution* module though is responsible to perform pseudonym resolution and it is used by PRA, PCA, LTCA as well as police authority. Finally, *dataBaseManagement* is an interface to accomplish all the database transactions. This interface is used by the entire modules within the package.

Chapter 5

Performance Evaluation

5.1 Introduction

The evaluation shows how the design and implementation satisfy the functional and security requirements of the project. Having done the design and implementation of the project, there exist some criteria to evaluate the artifact at hand. Such criteria shall consider different aspects of security within the domain of the secure VC system.

In this section, the evaluation criteria regarding VPKI will be stated. Later on, the experiments provides additional details. The experiments are performed in the experimentation lab, mentioned in chapter 4. The final results regarding the requirements and the goals are displayed. Furthermore, due to the achieved results, several optimizations will be identified in order to enhance the performance of the protocols.

5.2 Evaluation Criteria

The evaluation criteria for the master thesis project at hand are divided into two sections: evaluation in terms of functional requirements as well as evaluation in terms of security requirements. Both functional and security requirements, stated in chapter 2, shall be considered, verified and evaluated.

Performance, as one of the main criterion in every system, is not really a critical issue in this case while the correctness of the implementation and the efficiency are more significant. The system has to be evaluated to see if the secure vehicular communication system is as effective as a vehicular communication system without security features. Based on the *"principle of psychological acceptability"*[1], a secure system shall be as effective as the system without security features.

In this project, some standard libraries are being used such as: *OpenCA* and *OpenSSL*. This alludes that using open-source implementations is more reliable than using proprietary implementations. *OpenCA* is selected as the base implementation of this project, which is an open-source project and it has a high level of assurance in

[1] According to Bishop: *"The principle of psychological acceptability states that security mechanisms should not make the resource more difficult to access than if the security mechanisms were not present."* [4].

implementation and correctness. Furthermore, documentation is another criterion to see how the design and implementation have been done. Moreover, different UML diagrams are designed to clarify the protocol details while they demonstrate the steps of each protocol.[2] To prove the correctness of the protocols though, formal proof shall be used.

To evaluate the time, it is reasonable to calculate how long a signature operation takes to calculate the entire operations for a specific protocol. To evaluate how long does it take to perform a series of operations, it is feasible to use different standard machines. As long as it does not take a huge amount of time, it is reasonably good. Optimization also is not really a matter of discussion since such protocols are supposed to be performed once a week, for example, and it is not a problematic issue if it takes few more seconds.

Briefly speaking, the criteria to evaluate the artifact would be efficiency, effectiveness, usability, extendability and scalability for future usage; for example, the capability of customizing the key size, supporting different standard algorithms are some of the features that would be an advantage for future usage. To put it in another way, it is all about the proof of concept, meaning that a method or a set of methods and protocols are being used to show that to what extent they are feasible in practice.

In the next section, the experiments for this project is explained. There are some experiments which have been done to evaluate the protocols. Each of them will be mentioned in details.

5.3 Performance Evaluation, Experiments and Results

To evaluate the performance of the VPKI, there are some experiments to evaluate the correctness of the proposed protocols in terms of functional and security viewpoints. The main criterion to evaluate is the time. Several experiments are needed to see how long each phase takes to be accomplished. For each and every protocol, several experiments are tested with different parameters.

5.3.1 Obtaining Token from LTCA

Obtaining a token from LTCA is the first step to achieve pseudonym certificates. This is a simple query and response from the client to the server. To achieve a more accurate result, this experiments were done 20 times. Figure 5.1 demonstrates the time interval for different operations to obtain a token. Additionally, table 5.1 pinpoints the average time intervals for each operation. As illustrated, the entire operations in an experiment takes 100.75 millisecond. Preparing the request to obtain a token takes 4.95 milliseconds while the entire operations on the sever side takes 8.75 milliseconds to issue a token. The interval on the network communication is 83.6 millisecond.

5.3.2 Obtaining Pseudonyms from PCA

Figure 5.2 demonstrates the latency in seconds to obtain pseudonym certificates on the y-axis. On the x-axis, the number of pseudonym certificates is demonstrated. Size of each pseudonym certificate is 2 KB while size of each pseudonym private key is 5 KB.[3] As illustrated in the figure, the entire operations to obtain 1 pseudonym is 126.2 milliseconds.

[2] They are illustrated in Appendix.

[3] Each private key contains the corresponding public key.

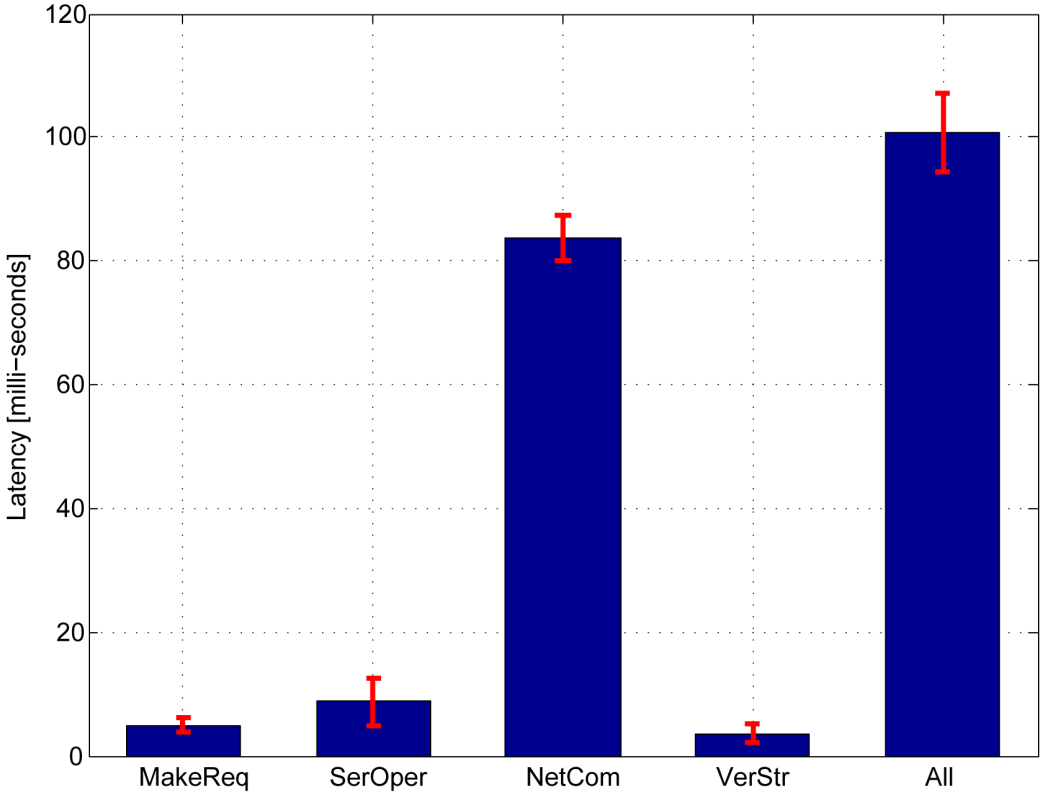

Figure 5.1: Time Intervals for Different Operations to Obtain a Token from LTCA

Table 5.1: *Time Intervals to Obtain a Token from LTCA*

Operation	**Time in Millisecond**
Preparing Token Request	4.95 ms
Issuing the Token (Server Side)	8.75 ms
Entire Communication	83.6 ms
Verification and Storage of the Token	3.65 ms
Entire Operations	100.75 ms

The time to obtain 10,000 pseudonyms is 165505.6 ms whereas the time to obtain 20,000 pseudonyms is 323673.4 ms. Table 5.2 pinpoints the time intervals to obtain different numbers of pseudonyms for each operation precisely.

Figure 5.3 illustrates the time interval to obtain 1 to 10 pseudonyms. The x-axis shows the number of pseudonyms whereas the y-axis demonstrates the latency in milliseconds.

Figure 5.4 shows the latency in seconds to obtain pseudonym certificates for different operations on the y-axis. On the x-axis, the number of pseudonym certificates is illustrated. The figure pinpoints the time intervals with different colors in order to reflect the most time consuming as well as the least time consuming operations. As clearly illustrated in the figure, the longest time interval are spent on the operations on the server side, to issue pseudonym certificates, and the operations on the client side, to prepare the

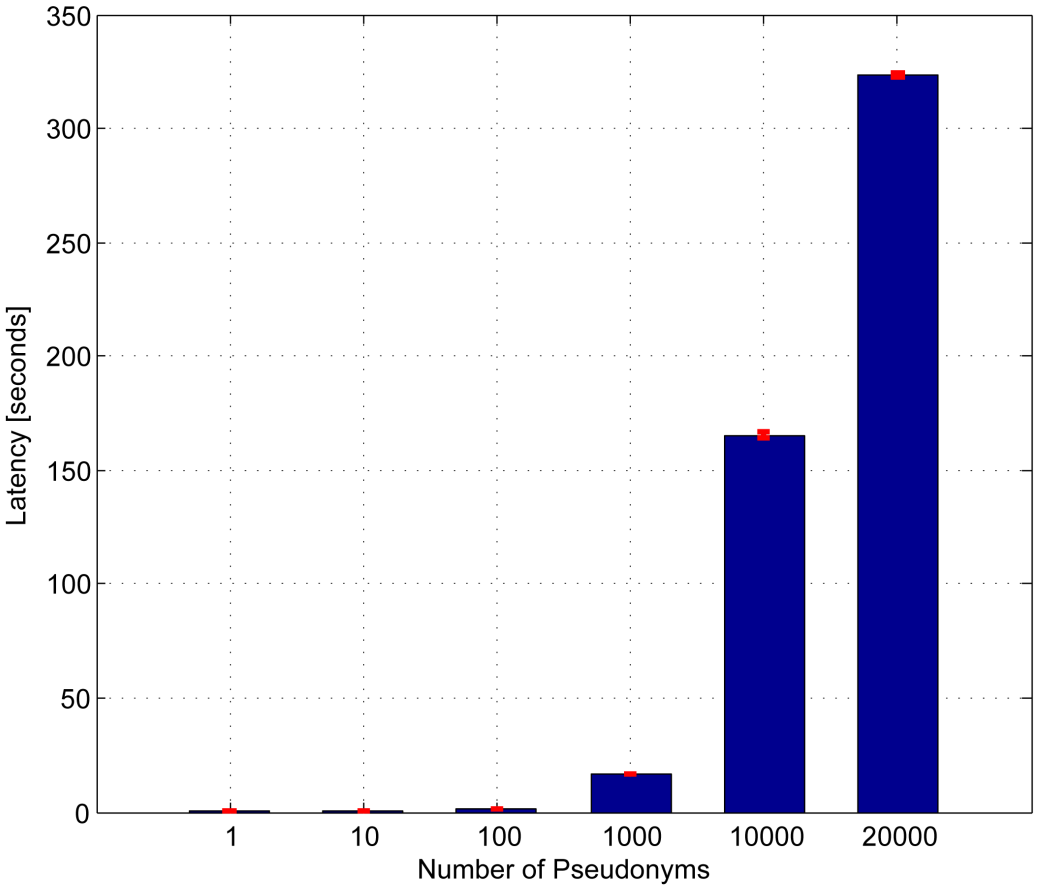

Figure 5.2: Time Intervals to Obtain Pseudonyms from PCA

request. Moreover, the shortest time interval is network communication, identified as yellow color in the figure.

Table 5.2: Time Intervals to Obtain Pseudonym Certificates

Operation	**Number of Pseudonyms**	**Time in Milliseconds**
Preparing the Request	1	10.5 ms
Entire Operations (Server Side)	1	18.2 ms
Entire Communication	1	69.7 ms
Verification and storage	1	9.7 ms
Entire Operations	1	126.2 ms
Preparing the Request	10	70.0 ms
Entire Operations (Server Side)	10	84.7 ms
Entire Communication	10	71.3 ms
Verification and storage	10	40.4 ms
		Continued on next page

Table 5.2 – continued from previous page

Operation	Number of Pseudonyms	Time in Milliseconds
Entire Operations	10	267.9 ms
Preparing the Request	100	461.9 ms
Entire Operations (Server Side)	100	805.9 ms
Entire Communication	100	125.8 ms
Verification and storage	100	367.4 ms
Entire Operations	100	1761.0 ms
Preparing the Request	1000	4246.8 ms
Entire Operations (Server Side)	1000	8027.6 ms
Entire Communication	1000	474.0 ms
Verification and storage	1000	3629.2 ms
Entire Operations	1000	16377.2 ms
Preparing the Request	10,000	43717.0 ms
Entire Operations (Server Side)	10,000	81207.7 ms
Entire Communication	10,000	3062.1 ms
Verification and storage	10,000	37519.0 ms
Entire Operations	10,000	165505.6 ms
Preparing the Request	20,000	83528.2 ms
Entire Operations (Server Side)	20,000	162837.6 ms
Entire Communication	20,000	5341.4 ms
Verification and storage	20,000	71965.9 ms
Entire Operations	20,000	323673.4 ms

Figure 5.5 shows the latency in seconds to perform different operations in order to obtain 20,000 pseudonym certificates on the y-axis. On the x-axis, different operations to issue 20,000 pseudonym certificates are represented. From left to right, they are identified as: *Key Generation*, *Making Request*, *Signature Verification (server side)*, *Generating Pseudonyms (server side)*, *Entire Server Operations*, *Network Communication*, *Verification and Storage (client side)* as well as the *entire operations*.

As represented below, generating 20,000 ECDSA keys takes 42612.7 milliseconds. The entire operations to prepare the request (including key generation) is 83528.2 milliseconds. On the server side, the time interval to verify the signature on 20,000 pseudonym requests is 68549.2 ms. Moreover, the time to generate pseudonyms is 70243.2 milliseconds while the entire operations on the server side (including certificate generation) takes 162837.6 milliseconds. The whole network communication interval is 5341.4 to send and receive data. Verifying as well as storing 20,000 pseudonym certificates on the client side is 71965.9 millisecond. The entire time interval to obtain 20,000 pseudonyms is 3236734 ms. As it is illustrated, the entire time to send and receive the entire information on the network is approximately 5341.4 milliseconds, which is not so high. It goes without saying that in reality, sending and receiving

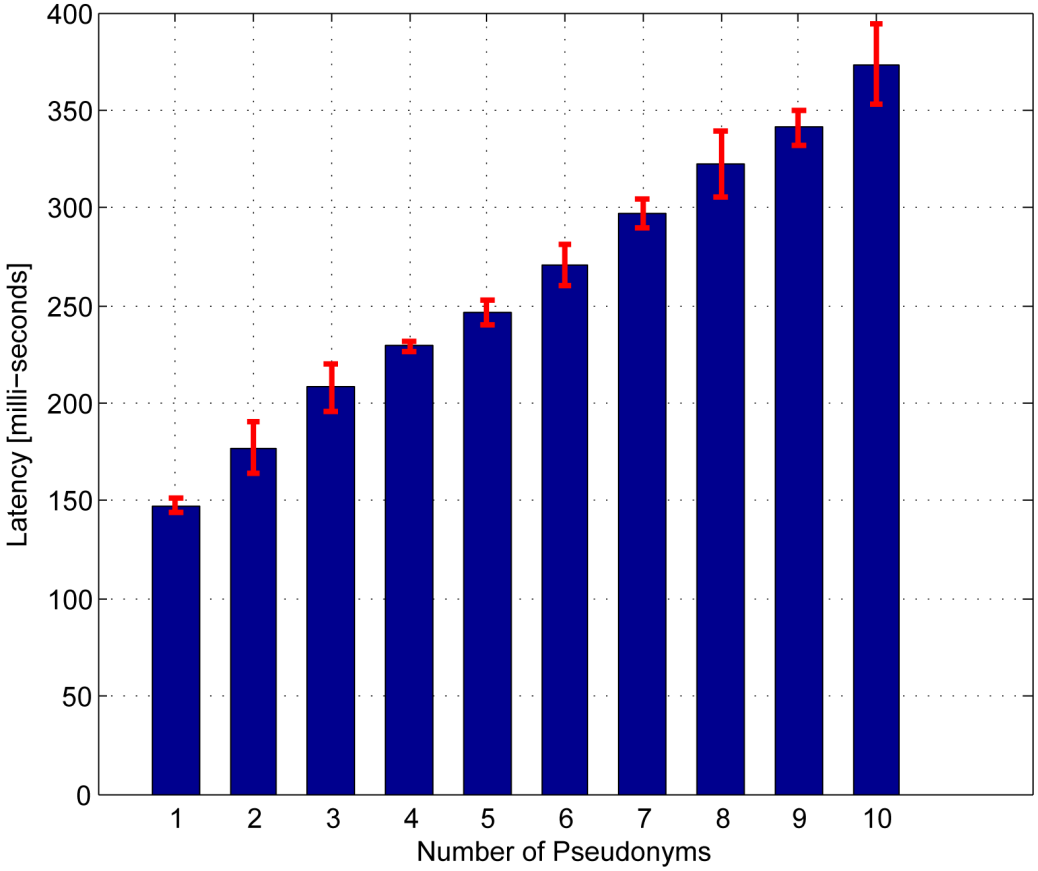

Figure 5.3: Time Intervals to Obtain 10 Consecutive Pseudonyms from PCA

information from a vehicle to the servers take more time. All in all, it does not affect the performance of the protocol as a whole since the time interval to transfer and receive the information is not so high.

Figure 5.6 draws attention to another perspective to obtain 20,000 pseudonym certificates. As shown in the figure, most of the time is spent to the operations on the server side including signature verifications as well as issuing pseudonym certificates. The second time consuming operation is preparing the request on the client side. The least time interval is the network communication, which is approximately 5341.4 milliseconds to send and receive data.

5.3.3 Obtaining CRL from PCA

Figure 5.7 draws the latency to obtain pseudonym CRL from PCA on the y-axis in milliseconds. On the x-axis, the number of revoked pseudonyms in the CRL is indicated. 10 different experiments have been done with different numbers of revoked pseudonyms in the database in order to evaluate the required time to receive CRL. As indicated in the figure, the entire time to obtain a pseudonym CRL, which contains 100,000 revoked pseudonyms, is 2756.8 millisecond. The precise time intervals to obtain CRL with different numbers of revoked pseudonyms is pointed out in table 5.3.

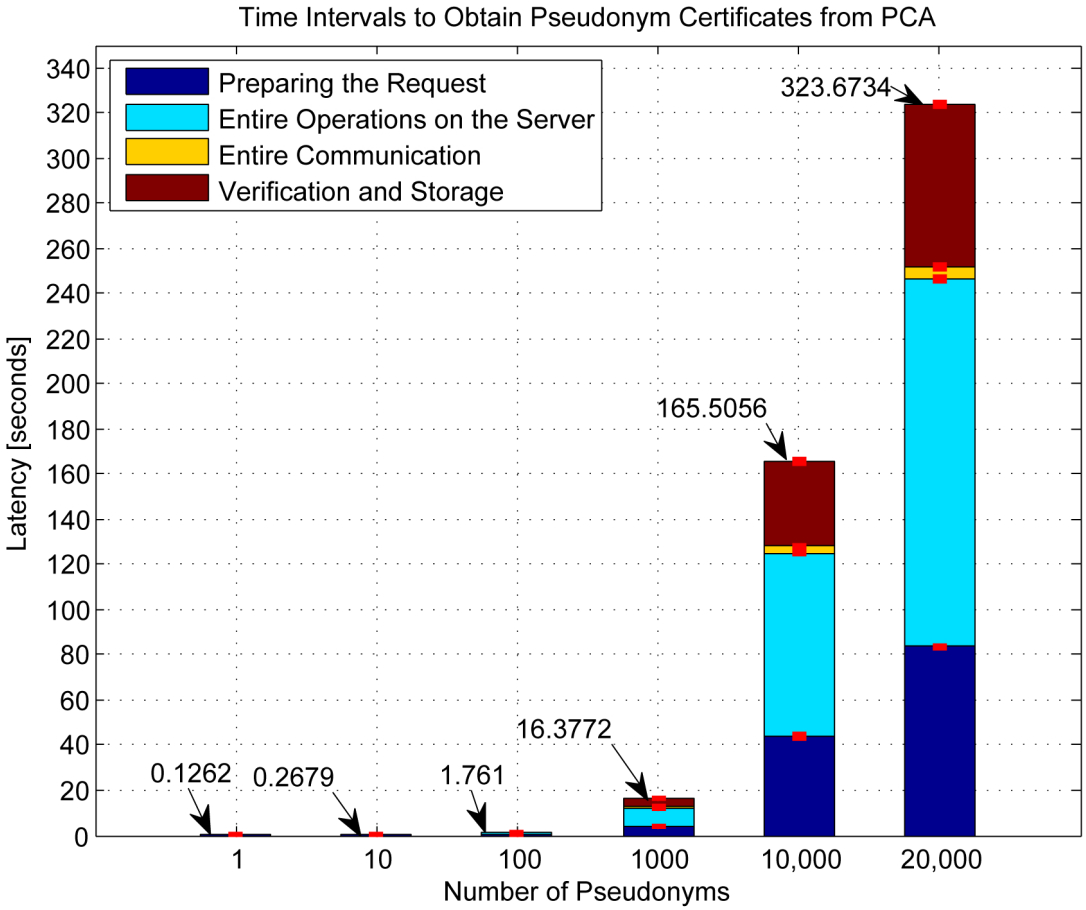

Figure 5.4: Time Intervals for Different Operations to Obtain Pseudonym Certificates

Figure 5.8 illustrates the latency to obtain pseudonym CRL for different operations in milliseconds on the y-axis. On the x-axis, the number of revoked pseudonyms in CRL is displayed. As indicated in the figure, the longest time interval to obtain pseudonym CRL is spent on the network communication, identified with yellow color. The time to perform the operations on the server and the time to verify and store the CRL on the client are almost identical. Preparing the request to obtain pseudonym CRL is the shortest time, shown with the blue color in the figure.

Table 5.3: Time Intervals to Obtain Pseudonym CRL

Operation	Revoked Pseudonyms in CRL	Time in Milliseconds
Preparing the Request	1	12.0 ms
Entire Operations (Server Side)	1	17.0 ms
Entire Communication	1	075.6 ms
Verification and Storage	1	5.5 ms
Entire Operations	1	110.2 ms

Continued on next page

Table 5.3 – continued from previous page

Operation	Revoked Pseudonyms in CRL	Time in Milliseconds
Preparing the Request	10	10.4 ms
Entire Operations (Server Side)	10	21.8 ms
Entire Communication	10	78.0 ms
Verification and Storage	10	5.1 ms
Entire Operations	10	115.6 ms
Preparing the Request	100	10.7 ms
Entire Operations (Server Side)	100	21.9 ms
Entire Communication	100	77.4 ms
Verification and Storage	100	5.4 ms
Entire Operations	100	115.3 ms
Preparing the Request	1000	11.6 ms
Entire Operations (Server Side)	1000	29.0 ms
Entire Communication	1000	85.2 ms
Verification and Storage	1000	11.2 ms
Entire Operations	1000	136.8 ms
Preparing the Request	10,000	13.2 ms
Entire Operations (Server Side)	10,000	100.6 ms
Entire Communication	10,000	199.1 ms
Verification and Storage	10,000	92.4 ms
Entire Operations	10,000	405.2 ms
Preparing the Request	20,000	11.7 ms
Entire Operations (Server Side)	20,000	171.2 ms
Entire Communication	20,000	331.9 ms
Verification and Storage	20,000	185.9 ms
Entire Operations	20,000	700.7 ms
Preparing the Request	100,000	8.8 ms
Entire Operations (Server Side)	100,000	769.2 ms
Entire Communication	100,000	1232.9 ms
Verification and Storage	100,000	745.7 ms
Entire Operations	100,000	2756.8 ms

Table 5.4 explicates the size of the pseudonym CRL with different revoked pseudonym certificates. As stated in the table, the size of a CRL with only one revoked pseudonym identifier is 778 bytes. The size of the CRL with 100,000 revoked pseudonyms is 6.48 MB.

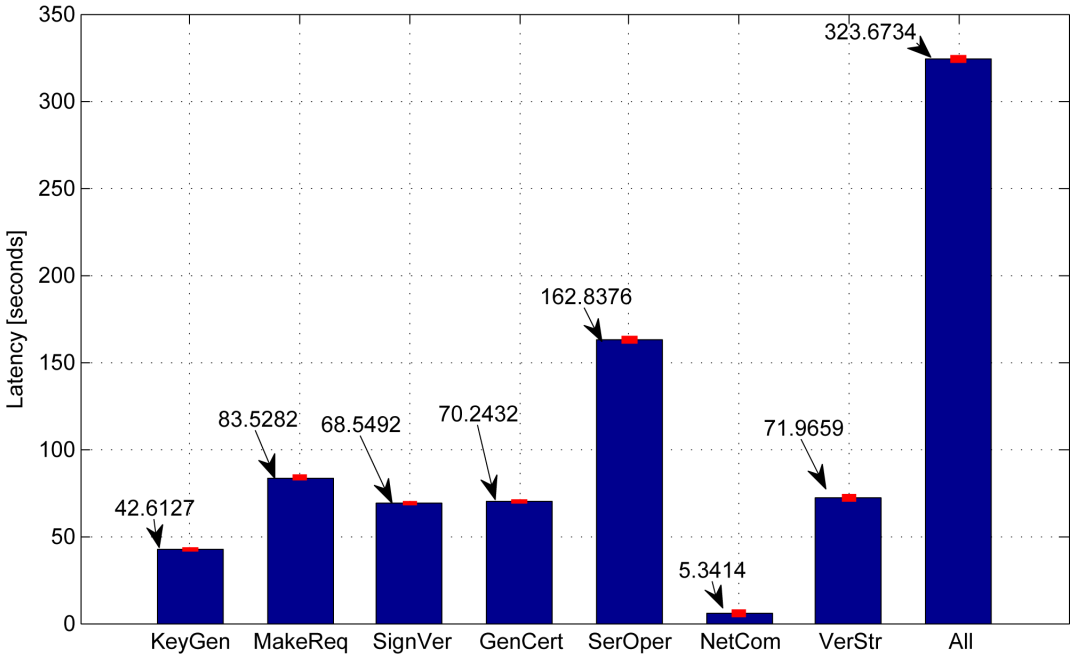

Figure 5.5: Time Intervals for Different Operations to Obtain 20,000 Pseudonym

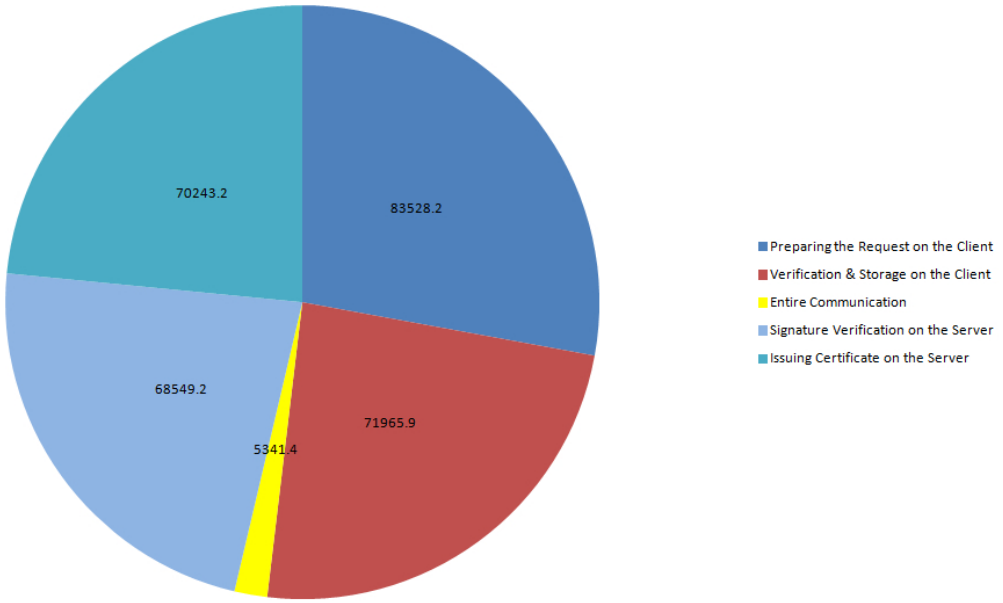

Figure 5.6: Percentage of Different Operations to Obtain 20,000 Pseudonym

5.3.4 Performing Pseudonym Resolution

Performing pseudonym resolution to figure out the real identity of a pseudonym consists of three query and response among the police authority, PCA, PCA and LTCA. In the first attempt of resolving a new pseudonym, PRA has to ask PCA as well as LTCA to obtain necessary information to resolve it. Afterwards for the upcoming requests, PCA is able to resolve it immediately since it has stored the information on its local database. This can

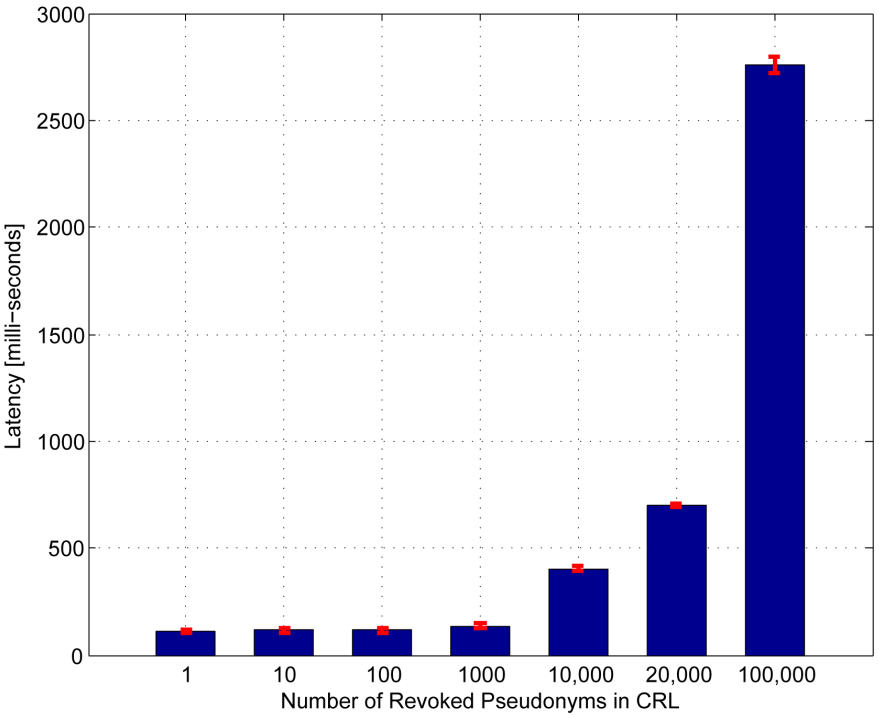

Figure 5.7: Time Intervals to Obtain Pseudonym CRL from PCA

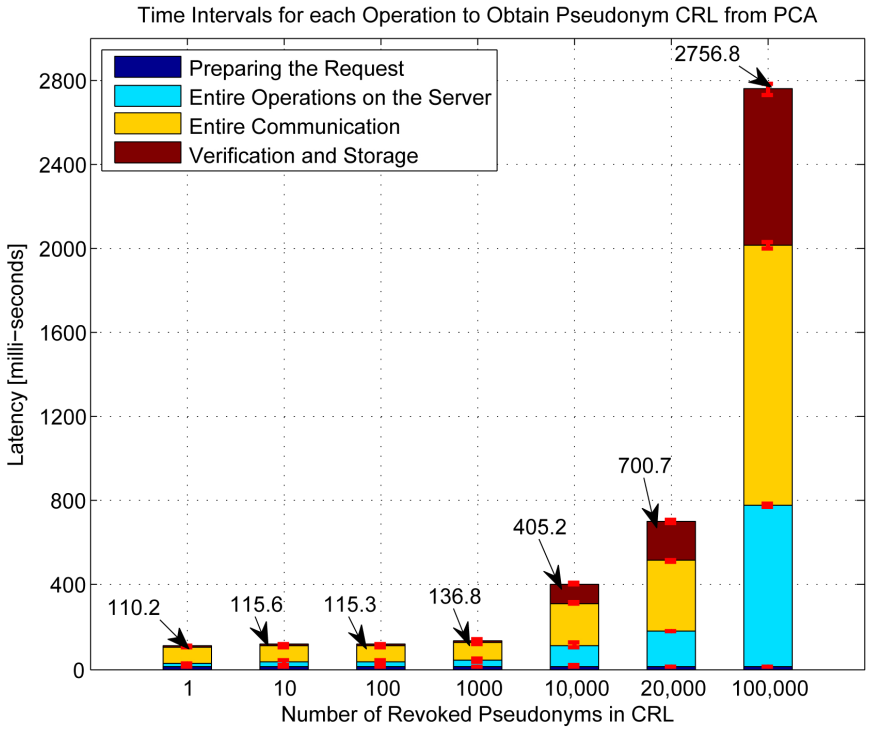

Figure 5.8: Time Intervals for Different Operations to Obtain Pseudonym CRL

yield in improving the performance, not to query for every request to PCA and LTCA. As a result, two different attempts have been done, as shown in table 5.5. On the first

Table 5.4: *Pseudonym CRL Size with Different Revoked Pseudonym Numbers*

Number of Revoked Pseudonyms in CRL	**Size in Bytes**
1	778 bytes (778 bytes)
10	1.36 KB (1,398 bytes)
100	7.33 KB (7,507 bytes)
1000	67.1 KB (68,723 bytes)
10,000	664 KB (680,718 bytes)
20,000	1.29 MB (1,360,714 bytes)
100,000	6.48 MB (6,800,715 bytes)

attempt, it takes 446 milliseconds to resolve the pseudonym while the second attempt to resolve the same pseudonym takes 104 milliseconds.

Table 5.5: *Performing Pseudonym Resolution to Resolve a Pseudonym*

	Time in Milliseconds
First Attempt	446 ms
Second Attempt	104 ms

No need to mention that all the experiments have been done in a lab using a computer, connected to the Internet via a LAN cable. When it comes to reality, each vehicle has to send and receive information via wireless. As a result, it could affect the time intervals. In the next chapter, the future works and the conclusion are explained to highlight the future direction of the project.

Chapter 6

Conclusion and Future Direction

6.1 Future Works

Providing a *PKI Trust Model in VANETs* is considered an essential requirement in order to extend the system as well. It could made up of RCA's hierarchies, cross-certification, or other mechanisms. Having fitted the system with a trust model hierarchy, vehicles from different realms are able to cross other realms while they can be authenticated and communicate with each other.

When it comes to scalability, *introducing a new PCA, LTCA and PRA* is one of the most important issues to extend the system. A mechanism should be introduced in order to elect a new authority for the system. This consists of requesting RCA for a new certificate, introducing the new authorities to the rest of the system, as well as obtaining other entities certificate, registered within the system.

With regard to traveling, *foreign certificates* are considered as a significant part of the system. *Roaming*, *issuing foreign certificates* as well as *revoking foreign certificates* were not discussed with great details since they are beyond the scope of this research. However, in terms of future work, this is an important issue since a vehicle can travel to different regions. Thereby, the system shall be furnished with a mechanism to support foreign pseudonym certificates. Moreover, keeping the system in a secure state is attributed to attaching the revoked foreign pseudonym certificates into the pseudonym CRL. It goes without saying that to provide such a system, authorities from different realms should cooperate with each other to authenticate the vehicles in order to issue tokens and pseudonym certificates.

When it comes to disseminating the latest version of pseudonym CRLs, each PCA publishes its local version of the CRL. A more comprehensive approach would be integrating all the revoked pseudonyms from different PCAs. The idea of integrating short-term CRLs from different PCAs would be an advantage since each vehicle can obtain all the pseudonym CRL from one PCA at once. Designing and implementing this feature require a new communication protocol, which has to be run once every few days among different PCAs.

In the current version of VPKI, each LTCA binds the token to a specific PCA to obtain pseudonym certificates. A more advanced approach to enhance the privacy of the vehicles would be issuing a generic token, which could be verified by any legitimate PCA. In this case, PCAs should cooperate with each other in order to identify if a token has been received pseudonym certificates. Therefore, LTCA cannot figure out which PCA will issue pseudonym certificates for a specific vehicle. This yields in increasing the level of privacy for the vehicles.

As illustrated in chapter 2, users' privacy is considered as the most significant security requirement within the VC system. However, according to the token format in section 3.3.1, an attacker can stage a *sybil attack* to masquerade as another legitimate vehicle. In the current version of the VPKI, it is assumed that each token never leaves the secure storage. However, to mitigate such a threat, appropriate countermeasures should be put in place, like: resource testing techniques, social networking approaches, radio testing as well as trusted certification.

Since PCA and LTCA never collaborate with each other in order to issue a pseudonym certificate, there has to be a mechanism to use the token once. In other words, PCAs should communicate with each other to check if a token has received pseudonym certificates. A protocol is necessary to be implemented among PCAs in order to verify that each token is used only once to obtain pseudonym certificates.

Performing reverse pseudonym resolution is another idea to figure out the pseudonym certificates for a long-term certificate. If a long-term certificate is revoked, an authority shall be able to do reverse pseudonym resolution in order to fetch the corresponding pseudonyms for that long-term certificate. The difference here is that PRA first queries LTCA to find Token-Id and in the second phase, it asks PCA to retrieve the corresponding pseudonym certificates for that Token.

As mentioned earlier, PRA is the responsible authority to perform pseudonym resolution. One of the PRA's responsibilities is to figuring out the pseudonyms belong to the same vehicle. The protocol should be designed and implemented to equip VPKI identifying related pseudonyms of a vehicle. Not to mention that in this case, the real identity of a pseudonym is not disclosed.

In the current version of VPKI, only one pseudonym is resolved at a time. In reality, multiple pseudonyms should be resolved by a request. Having implemented resolving multiple pseudonyms by each request, the performance is enhanced significantly since there is no need to resolve multiple pseudonyms in different requests.

From the implementation standpoint, using FastCGI instead of CGI boosts the performance and the efficiency. FastCGI is an advanced version of CGI that improves system availability as well as system security [42].

6.2 Summary

A vehicular public key infrastructure is designed and implemented using open-source libraries. OpenCA plays the main role in this infrastructure since all the required security functionalities for a PKI has been already implemented. OpenCA is equipped with extra security features in order to be customised for VANETs. These extra features comprises three protocols to obtain pseudonym certificates, pseudonym CRL, as well as performing pseudonym resolution. As demonstrated before, there are three main authorities involved in VPKI, identified as LTCA, PCA, and PRA. LTCA, Long-Term CA, is responsible to issue long-term certificates, long-term CRLs, as well as issuing

tokens for vehicles. This token is used in the process of issuing pseudonym certificates by PCA. Using the token, each vehicle is authenticated and authorized while its real identity is not disclosed. Moreover, PCA is responsible to issue pseudonym CRLs. The system is also provided with pseudonym resolution feature in order to identify the real identity behind a pseudonym certificate.

Having designed and implemented VPKI for VANETs, a great improvement is done in compare with similar projects, discussed in section 1.3. First and foremost, using a *token* to obtain pseudonym certificates helps to solve the problem of linkability among pseudonyms. Moreover, obtaining pseudonyms from any PCA enhances users' privacy since an LTCA cannot discover the issuer PCA. In other words, colluding PCA and LTCA is not an easy task to identify the real identity of a pseudonym; LTCA should collaborate with all the PCAs to identify the real identity since a vehicle can obtain pseudonyms from any PCA. Additionally, resolving a pseudonym requires 3 authorities, PRA, PCA and LTCA, to collaborate with each other. To put it in another way, users' privacy is protected not only from legal point of view, but also from technical perspective using separation of duties.

The results of experiences show that using the implemented version of the protocols to obtain pseudonym certificates as well as pseudonym CRLs is applicable to be used in VANETs. According to the achieved results, each vehicle might need to obtain 1000 pseudonyms each week and as a result, it might be taken approximately 11 seconds. It goes without saying that the experiments have been done on a computer, connected via a wired LAN to the Internet. To obtain a more precise result for the protocols, the same experiments should be done on a vehicle, driving in urban area. However, based on the results, the time intervals for network communications is not so high to affect the results as a whole.

Bibliography

[1] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, J.-P. Hubaux. *Secure Vehicular Communication Systems: Design and Architecture*. IEEE Commun. Mag., Nov. 2008.

[2] M. Emmelmann, B. Bochow, C. C. Kellum. *Automotive Standardization of Vehicle Networks, in Vehicular Networking: Automotive Applications and Beyond*. Ltd, Chichester, UK. doi: 10.1002/9780470661314. ch5, 2010.

[3] F. Kargl, P. Papadimitratos, L. Buttyan, M. Muter, E. Schoch, B. Wiedersheim, T.-V. Thong, G. Calandriello, A. Held, A. Kung, J.-P. Hubaux. *Secure Vehicular Communication Systems: Implementation, Performance, and Research Challenges*. IEEE Communications Magazine, 46(11), pages 110-118, 2008.

[4] M. Bishop. *Introduction to Computer Security*. 1^{st} ed. Addison-Wesley Professional, Prentice Hall PTR. 2004.

[5] T. Leinmuller, L. Buttyan, J.-P. Hubaux, F. Kargl, R. Kroh, P. Papadimitratos, M. Raya, E. Schoch. *Sevecom - Secure Vehicle Communication*. Proceedings of IST Mobile Summit, 2006.

[6] P. Papadimitratos, G. Mezzour, J.-P. Hubaux. *Certificate Revocation List Distribution in Vehicular Communication Systems*. ACM VANET, San Francisco, CA, 2008.

[7] P. Papadimitratos, V. Gligor, J.-P. Hubaux. *Securing Vehicular Communications - Assumptions, Requirements, and Principles*. In 4th Workshop on Embedded Security in Cars (ESCAR), pages 5-14, Berlin, Germany, 2006.

[8] B. Parno, A. Perrig. *Challenges in Securing Vehicular Networks*. In Workshop on Hot Topics in Networks (HotNets-IV), 2005.

[9] P. Golle, D. Greene, J. Staddon. *Detecting and Correcting Malicious Data in VANETs*. In Workshop on Vehicular Ad hoc Networks (VANET), 2004.

[10] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, A. Lloy. *Efficient and Robust Pseudonymous Authentication in VANET*. Proceedings of VANET, 2007.

[11] F. Schaub, F. Kargl, Ma. Zhendong, M. Weber. *V-Tokens for Conditional Pseudonymity in VANETs*. Wireless Communications and Networking Conference (WCNC), 2010 IEEE , vol., no., pp.1-6, 18-21 April 2010.

[12] M. Raya, P. Papadimitratos, J.-P. Hubaux. *Securing Vehicular Communications*. IEEE Wireless Communications Magazine, Special Issue on Inter-Vehicular Communications, Oct. 2006.

[13] PRESERVE Documentation, Development of Life-Cycle Management Components. *PRESERVE: Preparing Secure V2X Communication Systems*. Version 0.10.

[14] M. Bellare, D. Micciancio, B. Warinschi. *Foundations of Group Signatures: Formal Definition, Simplified Requirements and a Construction Based on Trapdoor Permutations*. In Eli Biham, editor, Advances in cryptology - EUROCRYPT 2003, volume 2656 of Lecture Notes in Computer Science, pages 614-629, Warsaw, Poland, Springer-Verlag, May 2003.

[15] M. Bellare, H. Shi, C. Zhang. *Foundations of Group Signatures: The Case of Dynamic Groups*. In CT-RSA'05, Lecture Notes in Computer Science, Springer-Verlag, 2004.

[16] P. Papadimitratos. *IoT Security - Part 1: Secure Vehicular Communication Systems*. Networked Systems Security (NSS), Lecture Notes, Slide #32, 2012. [online] Accessed on May 3, 2012. `http://www.ee.kth.se/~papadim/nss/nss.html`.

[17] Hevner, A.R. et al. *Design Science in Information Systems Research*. MIS Quarterly, 28(1), pp.75-105, 2004. [online] `http://www.jstor.org/stable/25148625`.

[18] P. Johannesson, E. Perjons. *A Design Science Primer*. [online] Accessed on April 3, 2012.

[19] P. Papadimitratos. *"On the Road" - Reflections on the Security of Vehicular Communication Systems*. International Conference on Vehicular Electronics and Safety, pp. 359-363, 2008.

[20] ETSI TS 102 867, European Telecommunications Standards Institute. *"Intelligent Transport Systems (ITS), Security, Stage 3 mapping for IEEE 1609.2"*. Version 2, Nov. 2011.

[21] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, A. Lioy. *On the Performance of Secure Vehicular Communication Systems*. IEEE Transactions on Dependable and Secure Computing, 2011.

[22] H. Baier, V. Karatsiolis. *Validity Models of Electronic Signatures and their Enforcement in Practice*. EuroPKI2009 - Sixth European Workshop on Public Key Services, Applications and Infrastructures, Sep. 2009.

[23] openssl.org. OpenSSL, Cryptography and SSL/TLS Toolkit. [Online] Accessed on 2012, April. `http://www.openssl.org/`.

[24] openca.org. OpenCA Guide. *OpenCA Guide for Versions 0.9.2+*. [Online] Accessed on 2012, April.

[25] mozilla.org. Mozilla, Open Source PKI Project. [Online] Accessed on 2012, April.

[26] A. Pfitzmann, M. Hansen. *Anonymity, Unlinkability, Unobservability, Pseudonymity, and Identity Management - A Consolidated Proposal for Terminology*. V0.25, Dec. 2005.

[27] A. Khalique, K. Singh, S. Sood. *Implementation of elliptic curve digital signature algorithm*. Int. Journal of Computer Applications, vol. 2, no. 2, pp. 21-27, 2010.

[28] G. McGraw. *Software Security: Building Security In*. Addison Wesley Professional, 2006.

[29] C. Kaufman, R. Perlman, M. Speciner. *Network Security: Private Communication in a Public World*. 2^{nd} ed. Prentice Hall, 2002.

[30] P. Papadimitratos, Z.J. Haas. *Secure Message Transmission in Mobile Ad Hoc Networks*. Elsevier Ad Hoc Networks J., Elsevier, vol. 1, no. 1, pp. 193-209, 2003.

[31] openldap.org. OpenLDAP Software. *OpenLDAP, Community Developed LDAP Software*. [Online] Accessed on 2012, April.

[32] openca.org. OpenCA Labs (Open-Source Security and Identity Management Solutions). [Online] Accessed on 2012, April.

[33] ospkibook.sourceforge.net. The OpenCA Project. *The Open-Source PKI Book: A Guide to PKIs and Open-Source Implementations*. [Online] Accessed on 2012, April.

[34] ejbca.sourceforge.net. *EJBCA PKI BY PRIMEKEY, Primekey Support, Development and Maintenance Services (The User Guide)*. [Online] Accessed on 2012, April.

[35] wiki.ejbca.org. EJBCA Wiki: EJBCA Open Source Enterprise PKI. [Online] Accessed on 2012, April.

[36] pyca.de. pyCA - X.509 CA (Software for Running a X.509/PKIX Certificate Authority). [Online] Accessed on 2012, April.

[37] sec.cs.kent.ac.uk. PERMIS (An Authorisation Infrastructure). [Online] Accessed on 2012, April.

[38] wikipedia.org. PERMIS. *PERMIS (PrivilEge and Role Management Infrastructure Standards)*. [Online] Accessed on 2012, April.

[39] datatracker.ietf.org. Public-Key Infrastructure (X.509) (pkix). [Online] Accessed on 2012, April.

[40] srg.cs.uiuc.edu. Oscar, a DSTC's PKI Prototype. [Online] Accessed on 2012, April.

[41] XMLRPC. A lightweight RPC library based on XML and HTTP. [Online] Accessed on 2012, September. `http://xmlrpc-c.sourceforge.net/`.

[42] FastCGI. [Online] Accessed on 2012, September. `http://www.fastcgi.com/drupal/`.

Appendices

.1 Protocol Design

.1.1 UML Diagram to Obtain Pseudonym Certificates

The description on the protocol is explained in chapter 3, in section 3.3. The list of notations, used in the protocol description, is also available in the list of Notations. Here, the detailed description on obtaining short-term certificates is described.

Obtaining pseudonym certificates comprises two phases. The first step is to get a token from LTCA to be used in the second phase. The second step is to obtain a limited number of pseudonym certificates from a PCA using that token. The idea is similar to SAML[1] ticket.

.1.1.1 Obtaining Token from LTCA

The first step is to ask an LTCA to receive a token. In the following, the procedures are explained in details:

Step #1: A vehicle does the following steps in order to obtain a token:

a) *Establish a secure TCP connection with LTCA (SSL, Mutual Authentication)*

b) *Encapsulate {REQ_TOKEN, Long-Term Cert. Length, Long-Term Cert., LTCA's ID, PCA's ID, Nonce, time-stamp}*

c) *Sign the value of {REQ_TOKEN, Long-Term Cert. Length, Long-Term Cert., LTCA's ID, PCA's ID, Nonce, TS} using vehicle's long-term private-key*

d) *Encrypt the entire message, {REQ_TOKEN, Long-Term Cert. Length, Long-Term Cert., LTCA's ID, PCA's ID, Nonce, TS, signed value in (c)} using* SK_{V-LTCA} *(from SSL)*

e) *Encapsulate the message*

f) *Send msg #1 to the LTCA*

(msg #1) Veh –> LTCA:

[1] Security Assertion Markup Language

SK_{V-LTCA}{REQ_TOKEN, Long-Term Cert. Length, Long-Term Cert., LTCA's ID, PCA's ID, Nonce, TS, σ_V^{LTC}(ENTIRE MESSAGE) }

Step #2: The LTCA accomplishes the following steps to issue a token:

a) *Decrypt the message and verify the integrity using SK_{V-LTCA} and IK_{V-LTCA} (from SSL)*

b) *Validate the vehicle's long-term certificate*

c) *Verify the signature on the value of {REQ_TOKEN, Long-Term Cert. Length, Long-Term Cert., LTCA's ID, PCA's ID, Nonce, TS} using vehicle's long-term public key*

d) *Verify the message freshness using time-stamp to be fresh and be within a threshold (time duration is several seconds)*

e) *Check LTCA's ID to be its own identifier*

f) *Check if this vehicle's LTC-Id has asked for a token recently. A threshold is applied and each vehicle can make a query once in the threshold.*

g) *If any error happens, send error-code and error message description to the vehicle and skip the rest. Otherwise, set error code to null:*

h) *Generate a new token as: Token {PSEUDONYM_CERT_TOKEN, Token-SerialNo, Token-Identifiable-Key, LTCA's ID, PCA's ID, Maximum number of Pseudonym Certificate, Token-StartTime, Token-LifeTime, Pseudonym-StartTime, Pseudonym-ExpiryTime, Signature}*

i) *Sign the token using LTCA's private key*

j) *Encrypt the token with the corresponding PCA's public key*

k) *Record and log all necessary information in the LTCA database including: {token-SerialNo, token-Identifiable-Key, vehicle Long-Term Cert. SerialNo, VehLTCert, PCA-Id, maxNoPseudonymCert, tokenStartTime, tokenLifeTime, pseudonymStartTime, pseudonymExpiryTime, nonce, timeStamp, usedToken}*

l) *Sign {RES_TOKEN, Token Size, Token, Max. No. of Pseudonyms, LTCA's ID, PCA's ID, Nonce+1, Time-Stamp, Error Code, Error Message Description} using LTCA's private key*

m) *Encrypt {RES_TOKEN, Token Size, Token, Max. No. of Pseudonyms, LTCA's ID, PCA's ID, Nonce+1, Time-Stamp, error-code, error message description, signed value in (l)} using SK_{V-LTCA} (from SSL)*

n) *Encapsulate the message*

o) *Send msg #2 to the Vehicle*

(msg #2) LTCA –> Veh:

SK_{V-LTCA}{RES_TOKEN, Token Size, Token, Max. No. of Pseudonyms, LTCA's ID, PCA's ID, Nonce+1, TS, Err. Code, EMD, σ_{LTCA}(ENTIRE MESSAGE)}

Step #3: Finally, the vehicle performs the steps below to fetch the token:

a) *Decrypt the message and verify the integrity using SK_{V-LTCA} and IK_{V-LTCA} (from SSL)*

b) *Verify the signature on the value of {RES_TOKEN, Token Size, Token, Max. No. of Pseudonyms, LTCA's ID, PCA's ID, Nonce+1, TS, Err. Code, EMD} using LTCA's public key*

c) *Check error-code and error message description*

d) *If error-code is not null, skip the rest. Otherwise:*

e) *Check the freshness and validity using time-stamp to be fresh and be within a threshold (time duration is several seconds)*

f) *Check nonce+1 to be the expected one*

g) *Store the token in HSM to be used in the next phase*

h) *Close the secure TCP connection (SSL)*

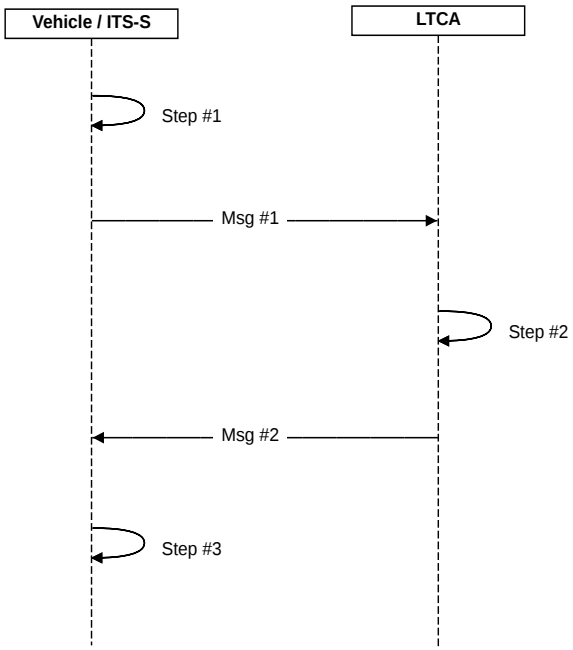

Figure 1: UML Diagram to Obtain a Token from LTCA

.1.1.2 Obtain Pseudonym Certificates from PCA

Step #1: A vehicle does the following steps in order to obtain a finite set of pseudonym certificates:

a) *Establish a secure TCP connection with PCA (SSL)*

b) *Generate a finite set of pseudonymous keys: K_V and k_V (using EC-DSA algorithm)*

c) *Encrypt {REQ_PSEU_CERT, Token Size, Token, LTCA's ID, PCA's ID, Location, PseuCert. No., $\{K_V^i\}_n$, Nonce, TS} using SK_{V-PCA} (from SSL)*

d) *Encapsulate the message*

e) *Send msg #1 to the PCA*

(msg #1) Veh –> PCA:

$$SK_{V-PCA}\{REQ\_PSEU\_CERT, Token\ Size, Token, LTCA's\ ID, PCA's\ ID, location, PseuCert.\ No., \{K_V^i\}, Nonce, TS\}$$

Step #2: The PCA performs the steps below:

a) *Decrypt the message and verify the integrity using SK_{V-PCA} and IK_{V-PCA} (from SSL)*

b) *Verify the message freshness using time-stamp to be fresh and be within a threshold (time duration is several seconds)*

c) *Check PCA's ID to be its own identifier*

d) *Decrypt the token using PCA's private key*

e) *Verify the signature on the token using the corresponding LTCA's public key*

f) *Verify the token's start-time, life-time and PCA's ID to be valid*

g) *Check the number of pseudonym certificates, which has to be at most equal to the maximum number of pseudonyms, identified in the Token*

h) *Check nonce to be fresh for this specific request to prevent against replay attack*

i) *Examining the short-term memory for (Token-ID, nonce)*

j) *If this value exists, skip the rest and notify the requester. Otherwise:*

k) *Add (Token-ID, nonce) to the short-term memory*

l) *Store (Token-ID, $\{K_V^i\}_n$) in the short-term memory (The set of pseudonym public keys that PCA may issue pseudonym certificate)*

m) *(Optional): Select some of the pseudonym public keys, and ask the vehicle to sign them in order to prove that the vehicle has the corresponding private keys (expensive operation)*

n) *If any error happens, send error-code and error message description to the vehicle and skip the rest. Otherwise:*

o) *Issue pseudonym certificates based on Start-Time and Life-Time, identified in the Token*

p) *Encapsulate the pseudonymous certificates, based on the pseudonym certificate format: {Pseu_Cert_SerialNo, PseuCert. Identifiable Key, PCA's ID, [t_i, $t_i+\Delta$], K_V^i, σ_{PCA}(ENTIRE PSEUDONYM)}*

q) *Record and log necessary information including: {pseudoCertSerialNo, tokenSerialNo, tokenIdentifiableKey, LTCA-Id, PCA-Id, Location, pseudonymCertNo, pseudonymPublicKey, pseudonymCert, nonce, timeStamp}*

r) *Sign the {RES_PSEU_CERT, LTCA's ID, PCA's ID, PseudCert. No., Pseudo-Cert. identified in (P), Nonce+1, time-stamp, error-code, error message description} using k_{PCA}, PCA's private key*

s) *Encrypt the {RES_PSEU_CERT, LTCA's ID, PCA's ID, PseudCert. No., Pseudo-Cert identified in (P), Nonce+1, Time-Stamp, error-code, error message description, signed value in (r)} using SK_{V-PCA} (from SSL)*

t) *Encapsulate a new message*

u) *Send msg #2 to the Vehicle*

(msg #2) PCA –> Veh:

$$SK_{V-PCA}\{\text{RES\_PSEU\_CERT, LTCA's ID, PCA's ID, PseudCert. No., } \{\text{Pseu\_Cert\_SerialNo, PseuCert. Identifiable Key, PCA's ID, } [t_i, t_i+\Delta], K_V^i, \sigma_{PCA}(\text{ENTIRE PSEUDONYM})\}, \text{Nonce+1, TS, Err. Code, EMD, } \sigma_{PCA}(\text{ENTIRE MESSAGE})\}$$

Step #3: Finally, the vehicle performs the steps below to retrieve the pseudonyms:

a) *Decrypt the message and verify the integrity using SK_{V-PCA} and IK_{V-PCA} (from SSL)*

b) *Verify the signature of the value of (RES_PSEU_CERT, LTCA's ID, PCA's ID, PseudCert. No., {Pseu_Cert_SerialNo, PseuCert. Identifiable Key, PCA's ID, [t_i, $t_i+\Delta$], K_V^i, σ_{PCA}(ENTIRE PSEUDONYM)}, Nonce+1, TS, Err. Code, EMD) using PCA's public key to prove the authenticity of the signer*

c) *Check time-stamp to be fresh and be within a threshold (time duration is several seconds)*

d) *Check error-code and error message description*

e) *If error-code is not null, skip the rest. Otherwise:*

f) *Check nonce+1 to be the expected one*

g) *Compare the signed pseudonym public keys, which shall be the same as the original ones*

h) *(Optional): Verify the signature of a subset of the short-term certificates*[2]

[2]This is an expensive operation. However, to be sure about the received short-term certificates, the vehicle can verify a subset of them. If an attacker manages to masquerade, his attempts will be recognized.

i) *Store the pseudonymous private keys in the HSM and the pseudonymous certificates in the corresponding storage, OBU*

j) *Encapsulate a new message to send the ACK*

k) *Send msg #3 to the PCA*

l) *Close the secure TCP connection (SSL)*

(msg #3) Veh –> PCA:

$$SK_{V-PCA}\{ACK\_OK,\ Nonce{+}2,\ TS,\ Err.\ Code,\ EMD\}$$

Step #4: PCA does the final step before closing the secure TCP socket:

a) *Decrypt the message and verify the integrity using SK_{V-PCA} and IK_{V-PCA} (from SSL)*

b) *If error-code is not null, check error message description. Otherwise, confirm the acknowledgment from the vehicle*

c) *Check time-stamp to be fresh and be within a threshold (time duration is several seconds)*

d) *Check nonce+2 to be the expected one*

e) *Close the secure TCP connection (SSL)*

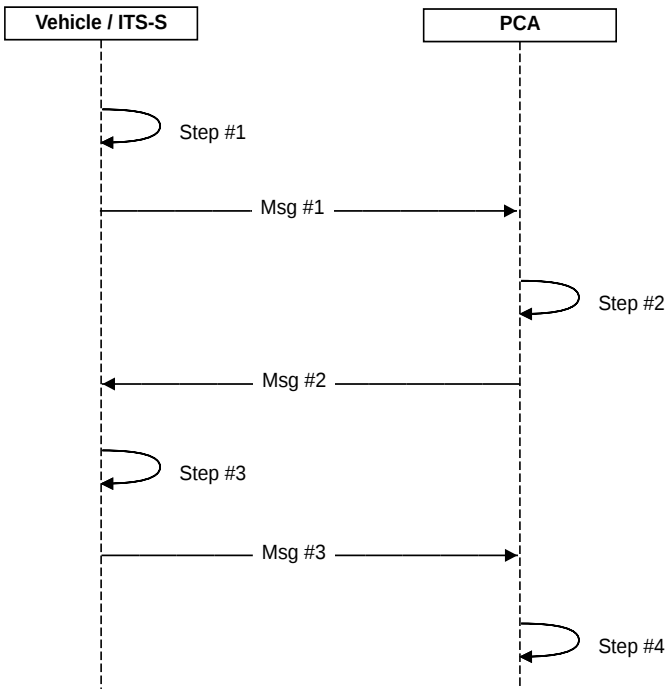

Figure 2: UML Diagram to Obtain Pseudonym Certificates

.1.2 UML Diagram to Obtain Short-Term CRL

The description on the protocol is explained in chapter 3, in section 3.4. The list of notations, used in the protocol description, is also available in the list of notations. Here, the detailed description of obtaining pseudonym CRL is pinpointed.

Step #1: A vehicle does the following steps in order to fetch the latest short-term CRL:

a) *Establish a TCP connection with PCA (Not necessarily an SSL connection)*

b) *Sign {REQ_PSEU_CRL, Current CRL Version, PCA's ID, Region-ID, PseuCert. Length, $Cert_{PCA}(K_V^i)$, Nonce, Time-Stamp} using the current valid pseudonym private key, k_V^i*

c) *Encapsulate the message*

d) *Send msg #1 to the PCA*

(msg #1) Veh –> PCA:

$$\{REQ\_PSEU\_CRL, Current\ CRL\ Version, PCA's\ ID, Region\text{-}ID, PseuCert.\ Length, Cert_{PCA}(K_V^i), Nonce, TS, \sigma_V^{k_v^i}(ENTIRE\ MESSAGE)\}$$

Step #2: The PCA accomplishes the following stages based on the request:

a) *Verify $Cert_{PCA}(K_V^i)$ and its life-time to be a valid pseudonym certificate*

b) *Verify the authenticity of the sender by validating the signature of the message on the value of (REQ_PSEU_CRL, Current CRL Version, PCA's ID, Region-ID, PseuCert. Length, $Cert_{PCA}(K_V^i)$, Nonce, Time-Stamp) using $Cert_{PCA}(K_V^i)$*

c) *Check time-stamp to be fresh and be within a threshold (time duration is several seconds)*

d) *Check nonce to be fresh for this specific request to prevent against replay attack*

e) *Examining the short-term memory for (Pseu-Cert-ID, nonce, time-stamp)*

f) *If this value exists, skip the rest and notify the requester. Otherwise, set error code to null:*

g) *Add (Pseu-Cert-ID, nonce, time-stamp) in the short-term memory*

h) *Logging the transaction by storing transaction information including: {(pseudoCert. SerialNo, CRL-Version, pseuCRL-SerialNo, PCA-Id, region-Id, time-stamp}*

i) *If any error happens, send error-code and error message description to the vehicle and skip the rest. Otherwise:*

j) *Fetch the latest short-term CRL (the current/local available short-term CRL)*

k) *Encapsulate {RES_PSEU_CRL, PCA's ID, CRL Size, Pseudonym CRL, Nonce+1, Time-Stamp, Error-Code, and Error Message Description}*

l) *Sign {RES_PSEU_CRL, PCA's ID, CRL Size, Pseudonym CRL, Nonce+1, Time-Stamp, Error-Code, and Error Message Description} using PCA's private key, k_V^i*

m) *Send msg #2 to the vehicle*

(msg #2) PCA –> Veh

{RES_PSEU_CRL, PCA's ID, CRL Size, Pseudonym CRL, Nonce+1, TS, Err. Code, EMD, σ_{PCA}*(ENTIRE MESSAGE)}*

Step #3: Finally, the vehicle performs these steps in order to obtain the latest short-term CRL:

a) *Check error-code and error message description*

b) *If error-code is not null, skip the rest. Otherwise:*

c) *Check time-stamp to be fresh and be within a threshold (time duration is several seconds)*

d) *Check nonce+1 to be the expected one*

e) *Extract the CRL*

f) *Verify the CRL signature using PCA's public key (PCA's ID is available in the message)*

g) *Compare the CRL version with the stored CRL version on the vehicle, to see if a newer version is released*

h) *If this is a newer version, store the CRL in the corresponding storage, OBU*

i) *Close the TCP connection*

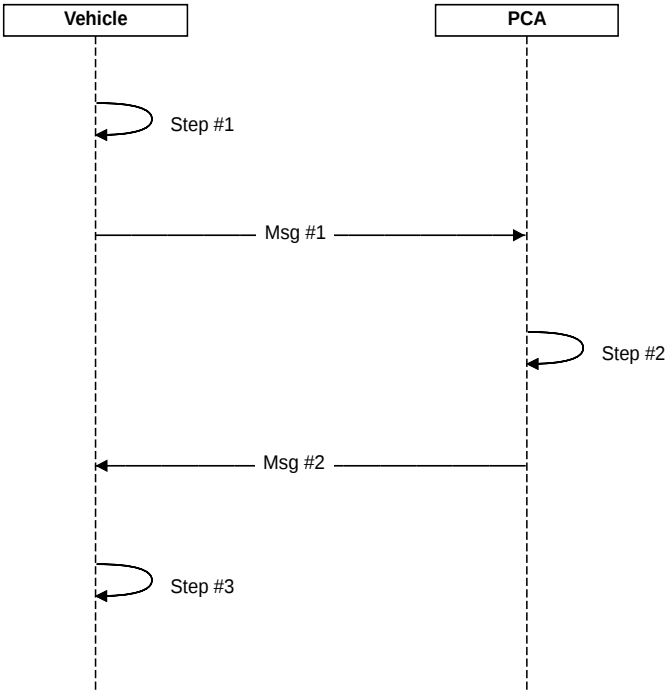

Figure 3: UML Diagram to Obtain Pseudonym CRL

.1.3 UML Diagram to Perform Pseudonym Resolution

The description on the protocol is explained in chapter 3, in section 3.5. The list of notations, used in the protocol description, is also available in list of notations. Here, the detailed description of performing pseudonym resolution is addressed.

This protocol is designed to identify the real identity of a pseudonym certificate, issued by a PCA. An authority such as police is able to query PRA (Pseudonym Resolution Authority) to fetch and retrieve the real identity. To do so, PRA first communicates with PCA to retrieve the token-ID and subsequently, it queries LTCA to identify the real identity of that token. It is assumed that a secure channel has been already established between PRA-PCA and PRA-LTCA.

Step #1: Police-authority accomplishes the steps below to obtain the real identity of a pseudonym:

a) *Establish a secure TCP connection with PRA (SSL, Mutual Authentication)*

b) *Sign {REQ_PSEU_RESOLUTION, PseuCert. Length, $Cert_{PCA}(K_v^i)$[3], Police's ID, PRA's ID, Nonce, TS} using the police-authority's private key, k_{Police}*

c) *Encrypt {REQ_PSEU_RESOLUTION, PseuCert. Length, $Cert_{PCA}(K_v^i)$, Police's ID, PRA's ID, Nonce, TS, the signed value in (b)} using $SK_{Police-PRA}$ (from SSL)*

d) *Encapsulate the message*

e) *Send msg #1 to the corresponding PRA*

(msg 1) Police-Authority –> PRA:

$$SK_{Police-PRA}\{REQ\_PSEU\_RESOLUTION, PseuCert.\ Length, Cert_{PCA}(K_v^i), Police's\ ID, PRA's\ ID, Nonce, TS, \sigma_{Police}(ENTIRE\ MESSAGE)\}$$

Step #2: PRA accomplishes the steps below to obtain the Token-ID:

a) *Decrypt the message and verify the integrity using $SK_{Police-PRA}$ and $IK_{Police-PRA}$ (from SSL)*

b) *Verify the signature on the value of (REQ_PSEU_RESOLUTION, PseuCert. Length, $Cert_{PCA}(K_v^i)$, Police's ID, PRA's ID, Nonce, Time-Stamp) using the police-authority's public key*

c) *Look up in its database if the real identity of the pseudonym is available. If it is available, go to step #6. Otherwise:*

d) *Sign {REQ_TOKEN_ID, PseuCert. Length, $Cert_{PCA}(K_v^i)$, PRA's ID, PCA's ID, Nonce, TS} using the PRA's private key, k_{PRA}*

e) *Encrypt {REQ_TOKEN_ID, PseuCert. Length, $Cert_{PCA}(K_v^i)$, PRA's ID, PCA's ID, Nonce, TS, the signed value in (d)} using $SK_{PRA-PCA}$ (from SSL)*

f) *Encapsulate the message*

[3]This is the pseudonym, which has to be resolved.

g) *Send msg #2 to the corresponding PCA*

(msg 2) PRA –> PCA:

$SK_{PRA-PCA}${*REQ_TOKEN_ID, PseuCert. Length,* $Cert_{PCA}(K_v^i)$*, PRA's ID, PCA's ID, Nonce, TS,* σ_{PRA}*(ENTIRE MESSAGE)*}

Step #3: PCA shall perform the following stages in order to retrieve the Token-ID:

a) *Decrypt the message and verify the integrity using* $SK_{PRA-PCA}$ *and* $IK_{PRA-PCA}$ *(from SSL)*

b) *Verify the signature on the value of (REQ_TOKEN_ID, PseuCert. Length,* $Cert_{PCA}(K_v^i)$*, PCA's ID, PRA's ID, Nonce, Time-Stamp) using* K_{PRA}

c) *Check time-stamp to be fresh and be within a threshold (time duration is several seconds)*

d) *Check* $Cert_{PCA}(K_v^i)$ *to see if the pseudonym is issued by a valid PCA*[4]

e) *Check nonce to be fresh for this specific request to prevent against replay attack*

f) *Examining the short-term memory for (Pseu-Cert-ID, nonce, time-stamp)*

g) *If this value exists, skip the rest and notify the requester. Otherwise:*

h) *Add (Pseu-Cert-ID, nonce, time-stamp) in the short-term memory*

i) *If any error happens, send error-code and error message description to the PRA and skip the rest. Otherwise:*

j) *Retrieve the corresponding Token-ID for this pseudonym from local PCA database*

k) *Set the error-code to null if the Token-ID exists and no error happens. Otherwise, set the error-code to the corresponding error code and the error message description to explain the reason in details.*

l) *Sign {RES_TOKEN_ID, PseuCert. SerialNo, Token SerialNo, Token Identifiable Key, PCA's ID, PRA's ID, LTCA's ID, Nonce+1, Time-Stamp, Error-Code, Error Message Description} using PCA's private key,* k_{PCA}

m) *Encrypt {RES_TOKEN_ID, PseuCert. SerialNo, Token SerialNo, Token Identifiable Key, PCA's ID, PRA's ID, LTCA's ID, Nonce+1, Time-Stamp, Error-Code, Error Message Description, the signed value in (l)} using* $SK_{PRA-PCA}$ *(from SSL)*

n) *Encapsulate the message*

o) *Send msg #3 to the PRA*

(msg 3) PCA –> PRA:

$SK_{PRA-PCA}${*RES_TOKEN_ID, PseuCert. SerialNo, Token SerialNo, Token Identifiable Key, PCA's ID, PRA's ID, LTCA's ID, Nonce+1, TS, Err. Code, EMD,* σ_{PCA}*(ENTIRE MESSAGE)*}

[4]There is no need for the pseudonym to be valid at the current time. It could be expired.

Step #4: PRA accomplishes the steps below to obtain the Token-ID:

a) *Decrypt the message and verify the integrity using $SK_{PRA-PCA}$ and $IK_{PRA-PCA}$ (from SSL)*

b) *Verify the signature on (RES_TOKEN_ID, PseuCert. SerialNo, Token SerialNo, Token Identifiable Key, PCA's ID, PRA's ID, Nonce+1, Time-Stamp, Error Code, Error Message Description) using the PCA's public key*

c) *Check error-code and error message description. If error code is not null, reply police authority based on error code and skip the rest. Otherwise, if error code is null:*

d) *Check time-stamp to be fresh and be within a threshold (time duration is several seconds)*

e) *Check nonce+1 to be the expected one*

f) *Check pseudonym certificate ID to be the expected one*

g) *Retrieve the Token-ID*

h) *Sign {REQ_REAL_IDENTITY, Token SerialNo, Token Identifiable Key, PRA's ID, LTCA's ID, Nonce, TS} using the PRA's private key, k_{PRA}*

i) *Encrypt {REQ_REAL_IDENTITY, Token SerialNo, Token Identifiable Key, PRA's ID, LTCA's ID, Nonce, TS, the signed value in (h)} using $SK_{PRA-LTCA}$ (from SSL)*

j) *Encapsulate the message*

k) *Send msg #4 to the corresponding LTCA*

(msg 4) PRA –> LTCA:

$$SK_{PRA-PCA}\{REQ\_REAL\_IDENTITY, Token\ SerialNo, Token\ Identifiable\ Key, PRA's\ ID, LTCA's\ ID, Nonce, TS, \sigma_{PRA}(ENTIRE\ MESSAGE)\}$$

Step #5: The LTCA shall perform the following stages in order to retrieve the real identity:

a) *Decrypt the message and verify the integrity using $SK_{PRA-LTCA}$ and $IK_{PRA-LTCA}$ (from SSL)*

b) *Verify the signature on the value of (REQ_REAL_IDENTITY, Token SerialNo, Token Identifiable Key, PRA's ID, LTCA's ID, Nonce, TS) using PRA's public key, K_{PRA}*

c) *Check time-stamp to be fresh and be within a threshold (time duration is several seconds)*

d) *Check nonce to be fresh for this specific request to prevent against replay attack*

e) *Verify Token-ID as a valid token*

f) *Examining the short-term memory for (Token-ID, nonce, time-stamp)*

g) *If this value exists, skip the rest and notify the requester. Otherwise:*

h) *Add (Token-ID, nonce, time-stamp) in the short-term memory*

i) *If any error happens, send error-code and error message description to the PRA and skip the rest. Otherwise, set error code to null:*

j) *Retrieve the corresponding real identity for this Token-ID from the local LTCA database*

k) *Sign {RES_REAL_IDENTITY, Token SerialNo, Veh. Long-Term Cert. SerialNo, VLTC Size, Veh. LTC, LTCA's ID, PRA's ID, Nonce+1, Time-Stamp, Error-code, Error Message Description} using LTCA's private key,* k_{LTCA}

l) *Encrypt {RES_REAL_IDENTITY, Token SerialNo, Veh. Long-Term Cert. SerialNo, VLTC Size, Veh. LTC, LTCA's ID, PRA's ID, Nonce+1, Time-Stamp, Error-Code, Error Message Description, the signed value in (k)} using* $SK_{PRA-LTCA}$ *(from SSL)*

m) *Log information in the database including: {Token-ID, Vehicle's real identity, LTCA's ID, PRA's ID, time-stamp, error code, error message description}*

n) *Encapsulate the message*

o) *Send msg #5 to the PRA*

(msg 5) LTCA –> PRA:

$SK_{PRA-LTCA}$*{RES_REAL_IDENTITY, Token SerialNo, Veh. Long-Term Cert. SerialNo, VLTC Size, Veh. LTC, LTCA's ID, PRA's ID, nonce+1, TS, Err. Code, EMD,* σ_{PCA}*(ENTIRE MESSAGE)}*

Step #6: The PRA does the steps below, to determine the real identity based on the token:

a) *Decrypt the message and verify the integrity using* $SK_{PRA-LTCA}$ *and* $IK_{PRA-LTCA}$ *(from SSL)*

b) *Verify the signature on the value of (RES_REAL_IDENTITY, Token SerialNo, Veh. Long-Term Cert. SerialNo, VLTC Size, Veh. LTC, LTCA's ID, PRA's ID, nonce+1, Time-Stamp, error code, error message description) using LTCA's public key,* K_{LTCA}

c) *Check error-code and error message description. If error code is not null, chec error message description for more information and notify the police authority about the error.*

d) *Check time-stamp to be fresh and be within a threshold (time duration is several seconds)*

e) *Check nonce+1 to be the expected one*

f) *If error code is null, retrieve the vehicle's real identity (VLTC's ID)*

g) *Log the information in the database for the later usage including: {pseuCert. SerialNo, token SerialNo, token Identifiable Key, vehicle LongTerm Cert. SerialNo, vehicle Long-Term Cert., pseuCertificate, Police-Id, PRA-Id, PCA-Id, LTCA-Id, time-stamp}*

h) *Sign {RES_PSEU_RESOLUTION, PseuCert Length,* $Cert_{PCA}(K_v^i)$*, Veh. Long-Term Cert. SerialNo, VLTC Size, Veh. Long-Term Cert., PRA's ID, LTCA's ID, PCA's ID, Police's ID, Nonce+1, Time-Stamp, Error-Code, Error Message Description)} using PRA's private key,* k_{PRA}

i) *Encrypt {RES_PSEU_RESOLUTION, PseuCert Length, $Cert_{PCA}(K_v^i)$, Veh. Long-Term Cert. SerialNo, VLTC Size, Veh. Long-Term Cert., PRA's ID, LTCA's ID, PCA's ID, Police's ID, Nonce+1, Time-Stamp, Error-Code, Error Message Description, the signed value in (h)} using $SK_{Police-PRA}$ (from SSL)*

j) *Encapsulate the message*

k) *Send msg #6 to the police-authority*

(msg 6) PRA –> Police-Authority:

$SK_{Police-PRA}$*{RES_PSEU_RESOLUTION, PseuCert Length,* $Cert_{PCA}(K_v^i)$*, Veh. Long-Term Cert. SerialNo, VLTC Size, Veh. Long-Term Cert., PRA's ID, LTCA's ID, PCA's ID, Police's ID, Nonce+1, TS, Err. Code, EMD,* σ_{PRA}*(ENTIRE MESSAGE)}*

Step #7: Finally, the police-authority does the steps below:

a) *Decrypt the message and verify the integrity using $SK_{Police-PRA}$ and $IK_{Police-PRA}$ (from SSL)*

b) *Verify the signature on the value of (RES_PSEU_RESOLUTION, PseuCert Length, $Cert_{PCA}(K_v^i)$, Veh. Long-Term Cert. SerialNo, VLTC Size, Veh. Long-Term Cert., PRA's ID, LTCA's ID, PCA's ID, Police's ID, Nonce+1, Time-Stamp, Error-Code, Error Message Description) using PRA's public key, K_{PRA}*

c) *Check time-stamp to be fresh and be within a threshold (time duration is several seconds)*

d) *Check nonce+1 to be the expected one*

e) *If error code is null, the Token-ID is resolved successfully*

f) *If error code is not null, check error message description for more information*

g) *Log the information in the database for the later usage including: {pseuCert. SerialNo, pseuCertificate, vehicle Long-Term Cert. SerialNo, vehicle Long-Term Certificate, Police-Id, PRA-Id, PCA-Id, LTCA-Id, time-stamp, error Info}*

h) *Encapsulate a new message to send the ACK*

i) *Send msg #7 to the PCA*

j) *Close the secure TCP connection (SSL)*

(msg #7) Police-Authority –> PRA:

$SK_{Police-PRA}$*{ACK_OK, Police's ID, PRA's ID, Nonce+2, TS, Err. Code, EMD,* σ_{PRA}*(ENTIRE MESSAGE)}*

Step #8: PRA does the final step before closing the secure TCP socket:

a) *Decrypt the message and verify the integrity using $SK_{Police-PRA}$ and $IK_{Police-PRA}$ (from SSL)*

b) *Verify the signature on the value of (ACK_OK, Police's ID, PRA's ID, Nonce+2, Time-Stamp, Error-Code, Error Message Description)*

c) *If error-code is not null, check error message description. Otherwise, confirm the acknowledgment from police-authority*

d) *Check time-stamp to be fresh and be within a threshold (time duration is several seconds)*

e) *Check nonce+2 to be the expected one*

f) *Close the secure TCP connection (SSL)*

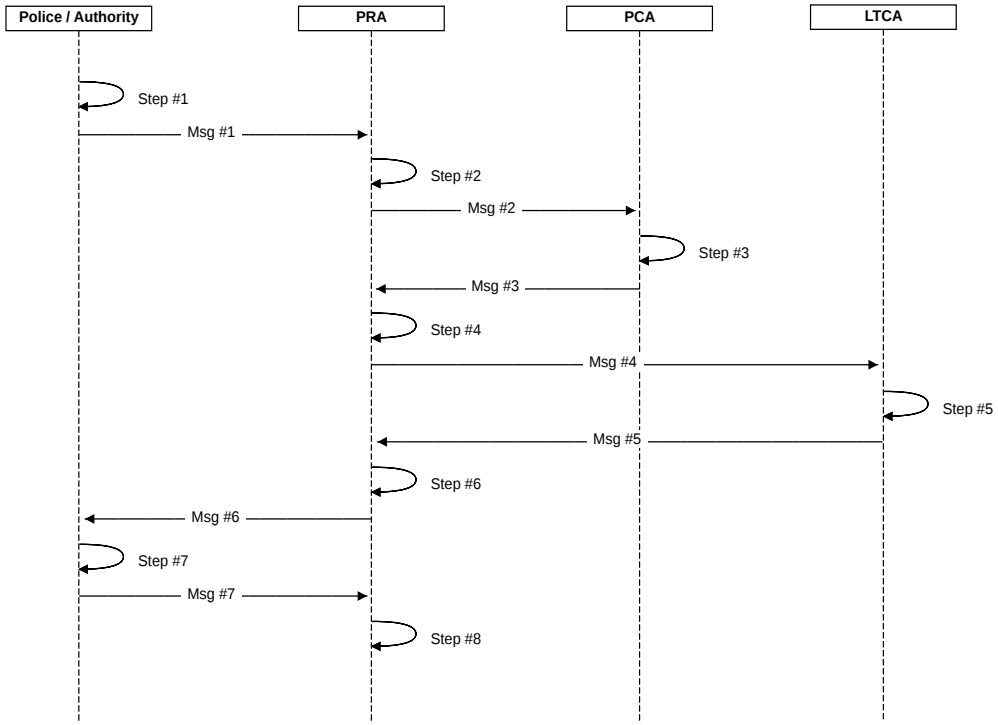

Figure 4: UML Diagram to Perform Pseudonym Resolution

The Royal Institute of Technology
School of Electrical Engineering
Osquldas väg 10
100 44
Stockholm
Sweden
Phone: +46 8 790 60 00
http://www.kth.se/ees